Theses and Dissertations   1. Thesis and Dissertation Collection, all items

1999-03

Automating the aviation command safety assessment survey as an Enterprise Information System (EIS)

Held, Jonathan S.; Mingo, Fred J.

Monterey, California: Naval Postgraduate School

http://hdl.handle.net/10945/13611

NAVAL POSTGRADUATE SCHOOL
MONTEREY, CALIFORNIA

THESIS

AUTOMATING THE AVIATION COMMAND SAFETY ASSESSMENT SURVEY AS AN ENTERPRISE INFORMATION SYSTEM (EIS)

by

Jonathan S. Held
Fred J. Mingo, Jr.

March 1999

Thesis Advisor:	C. Thomas Wu
Thesis Co-Advisor	Anthony Ciavarelli

Approved for public release; distribution is unlimited.

19990218186

REPORT DOCUMENTATION PAGE

Form Approved
OMB No. 0704-0188

1. AGENCY USE ONLY *(Leave blank)*	2. REPORT DATE March 1999	3. REPORT TYPE AND DATES COVERED Master's Thesis

4. TITLE AND SUBTITLE: AUTOMATING THE AVIATION COMMAND SAFETY ASSESSMENT SURVEY AS AN ENTERPRISE INFORMATION SYSTEM (EIS)	5. FUNDING NUMBERS
6. AUTHOR(S) Held, Jonathan S. and Mingo, Fred J. Jr.	

7. PERFORMING ORGANIZATION NAME(S) AND ADDRESS(ES) Naval Postgraduate School Monterey, CA 93943-5000	8. PERFORMING ORGANIZATION REPORT NUMBER

9. SPONSORING / MONITORING AGENCY NAME(S) AND ADDRESS(ES) Mr. Richard Healing Department of the Navy Safety and Survivability Bldg. 36 Washington Navy Yard 901 M Street, S.E. Washington, D.C. 20374-5028	10. SPONSORING / MONITORING AGENCY REPORT NUMBER

11. SUPPLEMENTARY NOTES

The views expressed in this thesis are those of the authors and do not reflect the official policy or position of the Department of Defense or the U.S. Government.

12a. DISTRIBUTION / AVAILABILITY STATEMENT Approved for public release; distribution is unlimited.	12b. DISTRIBUTION CODE

13. ABSTRACT *(maximum 200 words)*

The Aviation Command Safety Assessment (ACSA) is a questionnaire survey methodology developed to evaluate a Naval Aviation Command's safety climate, culture, and safety program effectiveness. This survey was a manual process first administered in the fall of 1996. The primary goal of this thesis is to design, develop, and test an Internet-based, prototype model for administering this survey using new technologies that allow automated survey submission and analysis.

The result of this thesis is a web site [http://spitfire.avsafety.nps.navy.mil] that adheres to a three-tier client/server architecture. The back-end SQL server database used to store survey information is accessed via front-end Java applets or Hypertext Markup Language (HTML) forms. Middleware components that complete the connection between client and server include Weblogic's Fastforward JDBC™ driver and Java servlets. The ACSA web site utilitizes many Internet technologies: Active Server Pages (ASP), HTML, Javascript, Active X, Secure Sockets Layer (SSL), CGI scripts, JDBC™, and Java applets and servlets. This thesis leads the reader through the research and development process describing how each of these technologies is used. Thorough review of this material is necessary for lifecycle support and future project revisions. Complete source code can be found in the appendices.

14. SUBJECT TERMS World Wide Web, Database, JDBC™, Java, Servlets, Active Server Pages, Hypertext Markup Language, Common Gateway Interface, Secure Sockets Layer, Internet Information Server, Windows NT Server, Aviation Command Safety Assessment	15. NUMBER OF PAGES 332
	16. PRICE CODE

17. SECURITY CLASSIFICATION OF REPORT Unclassified	18. SECURITY CLASSIFICATION OF THIS PAGE Unclassified	19. SECURITY CLASSIFICATION OF ABSTRACT Unclassified	20. LIMITATION OF ABSTRACT UL

NSN 7540-01-280-5500

Standard Form 298 (Rev. 2-89)
Prescribed by ANSI Std. 239-18
298-102

AUTOMATING THE AVIATION COMMAND SAFETY ASSESSMENT SURVEY AS AN ENTERPRISE INFORMATION SYSTEM (EIS)

Jonathan S. Held
Lieutenant, United States Navy
B.A., University of Pennsylvania, 1993

Fred J. Mingo, Jr.
Commander, United States Naval Reserve
B.Phil., Miami University, 1979

Submitted in partial fulfillment of the
requirements for the degree of

MASTER OF SCIENCE IN COMPUTER SCIENCE

from the

NAVAL POSTGRADUATE SCHOOL
March 1999

Authors: _____
Jonathan S. Held

_____
Fred J. Mingo, Jr.

Approved by: _____
C. Thomas Wu, Thesis Advisor

_____
Anthony Ciavarelli, Thesis Co-Advisor

_____
Dan Boger, Chairman
Department of Computer Science

ABSTRACT

The Aviation Command Safety Assessment (ACSA) is a questionnaire survey methodology developed to evaluate a Naval Aviation Command's safety climate, culture, and safety program effectiveness. This survey was a manual process first administered in the fall of 1996. The primary goal of this thesis is to design, develop, and test an Internet-based, prototype model for administering this survey using new technologies that allow automated survey submission and analysis.

The result of this thesis is a web site [http://spitfire.avsafety.nps.navy.mil] that adheres to a three-tier client/server architecture. The back-end SQL server database used to store survey information is accessed via front-end Java applets or Hypertext Markup Language (HTML) forms. Middleware components that complete the connection between client and server include Weblogic's Fastforward JDBC™ driver and Java servlets. The ACSA web site utilitizes many Internet technologies: Active Server Pages (ASP), HTML, Javascript, Active X, Secure Sockets Layer (SSL), CGI scripts, JDBC™, and Java applets and servlets. This thesis leads the reader through the research and development process describing how each of these technologies is used. Thorough review of this material is necessary for lifecycle support and future project revisions. Complete source code can be found in the appendices.

TABLE OF CONTENTS

LIST OF FIGURES

LIST OF TABLES

ACKNOWLEDGEMENT/DEDICATIONS

To Barbara, Alexander, Lukas, and Trevor: Without your support, this never would have been possible.

- Fred Mingo

To my Dad: I never thought I'd follow in your footsteps, but I can't think of a better person I'd like to follow than you.

To Lisa and Jacob: The light is at the end of the tunnel. Thanks for your patience and perseverance.

- Jonathan Held

I. INTRODUCTION

A.   PURPOSE

The purpose of this thesis is to design, develop, and test a prototype system that supports the Aviation Command Safety Assessment (ACSA) survey process. Our goal is to replace the manual, labor-intensive efforts that were previously required to administer this survey with an automated, less costly alternative.

The ACSA survey is part of an analytical methodology that was developed to help evaluate a naval aviation command's safety program effectiveness. In support of this program, our research will aim to examine several new and current web-based technologies, protocols, and standards for building Internet-based, client/server applications. Our efforts will culminate with the implementation of a prototype system that allows clients to remotely complete and submit the survey to a Structured Query Language (SQL) 6.5 database through the information medium that is the Internet.

Building such a system serves two primary purposes:

- It results in a reduction of the processing overhead that was formerly involved in the administration of the ACSA survey. Project managers will no longer have to mail surveys, tally results, archive data, or enter results into spreadsheet applications.

- It makes the results of the survey accessible in a timelier manner to the customers that have a critical need for the information. Additionally, these results are updated dynamically and subject to a more exhaustive set of queries than what was possible apriori.

By integrating existing Commercial Off-the-Shelf (COTS) products with solutions that are available free of charge, our prototype system costs very little. Hardware equipment was by far the biggest, single expense incurred, but because the cost of computers has been drastically reduced over the last year, it is becoming even cheaper to procure the equipment that is needed. The current prototype system uses a 300 MHz Pentium II running Windows New Technology (NT) Server 4.0 and Microsoft's Internet Information Server (IIS) 4.0. This configuration adequately covers the needs of the prototype system, and brought the total development cost to slightly less than $5,000.

1

B.   BACKGROUND

The aircraft mishap rate has declined substantially over the past fifty years of Naval Aviation. Just twenty years ago, the U.S. Navy aircraft mishap rate stood at about ten aircraft losses for every 100,000 flight hours. The loss rate has steadily dropped since that time to a present level of around two aircraft losses for every 100,000 flight hours. But while the mishap rate has declined appreciably, the number of aircraft losses due to aircrew factors, or human error, has stayed relatively constant.

About 60% of the U.S. Navy and U.S. Marine Corps "class A" aircraft mishaps (those resulting in death, permanent disability, or loss of $1 million or more) are caused by aircrew errors. In the spring of 1996, naval aviation experienced a series of highly publicized aircrew error mishaps. Many of these mishaps became the focus of national media attention and even congressional investigation. In an effort to proactively address these mishaps, Commander Naval Air Pacific (COMNAVAIRPAC) chartered a Human Factors Quality Management Board (QMB) to explore ways in lowering the naval aviation mishap rate.

While much attention has been paid to understanding human error in the cockpit, up until this time there was little research concerning how organizational factors affected crew performance and safety [Ref. 1]. The QMB recognized a squadron's "safety climate" was a critical link in the mishap chain-of-events, but there was no data supporting these opinions. Dr. Anthony Ciavarelli, Naval Aviation Safety School, was appointed as the principal investigator for measuring command safety climate throughout naval aviation as a whole. Specifically, his mission was to provide the QMB with feedback on issues concerning command climate, morale of personnel, workload and resource availability, estimated success of ongoing safety interventions, and other factors related to safely managing Naval flight operations.

Dr. Ciavarelli developed the ACSA questionnaire to evaluate a command's safety climate, culture, and safety program effectiveness. This 61-item questionnaire was administered by mail to 69 randomly selected navy and marine squadrons in the fall of 1996. The survey sample respondents were limited to flight-qualified naval aviators,

from the commissioned officer ranks only. Additionally, these participants were assured that their responses were anonymous and would remain that way, and that all results concerning squadron ratings would be held in strictest confidence. The questionnaire was constructed using a Likert type seven point rating scale with verbal anchors, as follows: (1) Strongly Disagree, (2) Moderately Disagree, (3) Disagree, (4) Neutral, (5) Agree, (6) Moderately Agree, (7) Strongly Agree. A "Not Applicable" category was later added to the questionnaire so respondents could refrain from answering any question which was not relevant, or that they were not qualified to answer. In the end, a total of 67 units (97%) responded to this survey, for a total of 1254 individual survey respondents. The results of this initial survey provided invaluable insight into the safety climate of squadrons, wings, Type Commanders, and naval aviation as a whole.

Today, the Human Factors QMB is eager to update this previous study with one that is identical in scope, but can be administered in a more timely manner and require less processing overhead. Automating the ACSA will accomplish this, and it will allow the QMB to quickly and efficiently survey and simultaneously evaluate communities throughout naval aviation. Results from the application of the automated ACSA will be used as they were before: to improve the safety process at the local command level, and to encourage overall process improvements in all naval aviation units. Such information regarding command safety effectiveness is an important step in accomplishing the U.S. Navy's long-term goal to dramatically reduce the number of aircraft mishaps and loss of life.

C.   THESIS ORGANIZATION

This thesis is organized into the following chapters:

- Chapter I: Introduction. This chapter provides an introduction for the thesis and the reason for research in the specific topic area.

- Chapter II: JDBC™. This chapter outlines how JDBC is used in the ACSA project. JDBC™ is the primary means of communicating with Relational Database Management Systems (RDBMS). This chapter describes how the ACSA project uses Java™ applets and database drivers for communicating with remote data sources, what restrictions are imposed upon these applets, minimum client requirements for using this system, as well as common security considerations.

- Chapter III: Java™ Servlets. Servlets are a Common Gateway Interface (CGI) substitute that allow development of server applications that can be run by remote clients. Servlets are far safer than CGI because of the Java™ programming language, and how servlets are used is the subject of this chapter.

- Chapter IV: Microsoft Active Server Pages (ASP). ASP is an integral part of the Active Platform, Microsoft's core Internet Strategy. ASP is actually a component that is installed on top of Microsoft's Internet Information Server (IIS) and processes files that have a ".asp" extension. Once the file is processed, IIS returns the output to the client's web browser. ASP applications complement web development and tools such as Visual Interdev facilitate the creation of ASP database applications. An overview of ASP, and how it was used in the ACSA project, is detailed in this chapter.

- Chapter V: Database Design and System Operation. This chapter discusses the database products and applications employed for the ACSA project. Topics include a discussion on design decisions made during the development of the ACSA database, the use of Microsoft Access and the upsizing utility, and how Microsoft's NT Server 4.0 and IIS 4.0 are incorporated and configured for this project.

- Chapter VI: Conclusions. Here, we present an overall assessment of the automated ACSA process. We look at performance issues and make recommendations for future research and application.

D.   SOME ASSUMPTIONS

In order to make the ACSA site accessible to as many clients as possible, we developed software code to accommodate versions 4.0+ of Netscape's Navigator and Microsoft's Internet Explorer web browsers. The reader should know how to use these products. Additionally, he or she should have a basic understanding of programming language principles and how to use Structured Query Language (SQL), the specification used to talk to databases. Many of the chapters reference Java™ code and explain what portions of the code are used for, but the full context of what is going on and why can only be gleaned through meticulous examination of the complete source code. Aside from these minimum requirements, we have gone to great length to provide detailed descriptions of how a system is properly configured and how you use the technologies we employed. Once you have read this material in its entirety, you should feel comfortable enough to do what we have done and implement your own Internet-accessible database.

E. SIDENOTES

On March 3, 1997, the U.S. Navy promulgated a policy called Information Technology for the 21st Century, or IT-21, which stipulated the minimum hardware and software requirements for Department of the Navy (DON) Information Systems (IS). IT-21 was intended to standardize DON-wide IS development, and as such, we have adhered to the policy's guidelines. Each component of the prototype system currently meets the IT-21 specification. To learn more about IT-21, and what requirements it places on developers, see Appendix A.

II. JDBC™

A.   BACKGROUND

Prior to the advent of JDBC™, programmers had few choices as to how they could get their applications to communicate with Relational Database Management Systems (RDBMS). By far the most common solution was to use Microsoft's Open Database Connectivity (ODBC) Application Programming Interface (API). This API allowed developers to use a standard set of Structured Query Language (SQL) statements in any application, and it would translate commands into a syntax that database servers could understand. But for all ODBC allowed you to do, it had a number of attributes that made it difficult to effectively use:

- ODBC was written exclusively in the C programming language, so there was no concept of objects or methods. The logical organization that is indicative of Object Oriented Programming (OOP) was nowhere to be found, resulting in a great deal of frustration when you were trying to find the right procedure or function to call [Ref. 2].

- The API was extremely large, hard to follow, and required an intimate knowledge of the entire API if you wanted to make it work for you [Ref. 2].

In March, 1996, Sun Microsystem's stepped in as a database communication provider. It was at this time that the company introduced its JDBC™ API, which is a set of Java™ objects and methods used to execute SQL statements on RDBMS. JDBC™ quickly caught on as an alternative solution for interacting with data warehouses because it was so easy to use. It presented the developer with a small object model that you could quickly learn and then use in Java™ applications. Because it embraced the OOP paradigm, it had a greater degree of logical organization than ODBC [Ref. 2].

For the last several years, Java™ and JDBC™ have proven their worth as critical components in successfully expanding the presence and potential of network-based client/server database computing. It is hard to navigate the Internet today and not see both of these components at work. And because JDBC™ is part of the Java™ Virtual Machine (JVM) specification, its continued use is altogether guaranteed.

B.   JDBC™ IN THE ACSA PROJECT

JDBC™ is not part of the core Java™ programming language but is included for use in the package *java.sql*. Because it is part of the JVM, web browsers that provide applet support also natively allow you to use the objects and their methods from this package. Unfortunately, there is one more, costly element that is required in order to use JDBC™ - the driver. A driver is a program that interprets your JDBC™ calls and converts them into a syntax that can be used for two-way communication between your program and a database server. Once the driver is properly installed, you are ready to begin. Discussion on the driver we chose, its comparative cost to other commercial drivers, its installation, and the different types of drivers available are presented later in this chapter.

The ACSA web site, located at [http://spitfire.avsafety.nps.navy.mil] has two Java™ applets that use a driver and JDBC™ to communicate with RDBMS:

- The *Survey* applet allows the client to download a Graphical User Interface (GUI) application that has ten biographical and 61 survey questions. The applet ensures that all responses are answered correctly and completely, and it confirms the user is authorized to input information before forwarding the data for input to the backend SQL Server 6.5 database.

- The *Results* applet is also a GUI application that allows the client to query the results of all survey submissions. This applet is extremely flexible because it allows the user to narrow his query criteria by specifying the question number and any combination of community, service, status, or location. Results are presented in a colorful histogram which is based on the range of responses.

No matter which applet is being used, the procedure for communicating with the back-end data warehouse is the same. When the client wants to input or retrieve information, he must wait long enough for the driver to be loaded. This temporarily halts program execution and depending upon the client's network connection speed, can take varying amounts of time. Once the driver is loaded, the application can begin processing its JDBC™ calls (see Figure 2.1).

8
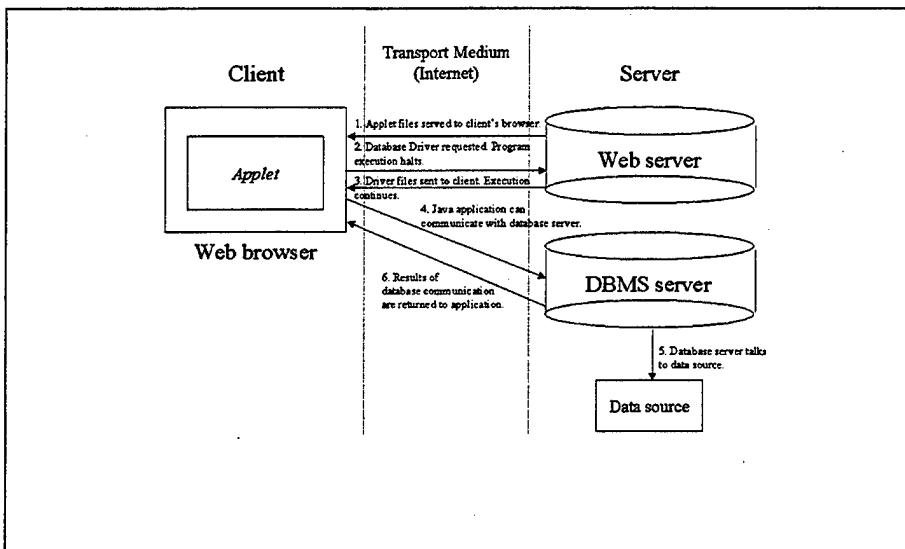
Figure 2.1. Applet communication with a remote data source.

The ACSA project also allows the client to interact with a remote data source using an HTML form. This form contains the same questions as the *Survey* applet and was developed primarily as a secondary means of submitting information should the client not have the computing resources necessary for using applets. For information on how the HTML front-end works and its interaction with a Java™ technology known as servlets, the reader should refer to Chapter III. The client requirements for using the Java™ applets are listed on the home page of the ACSA web site.

As a final sidenote, future developers of the ACSA project should be aware that the survey database is named *Survey2_DB*. There is also a second database, *Num_List_DB*, that is used to authenticate survey users. Both databases were initially designed using Microsoft Access, then "upsized" to a SQL 6.5 database using an Access plug-in. For more information on how to upsize a database, and a brief overview of how to use the SQL Server 6.5 *Enterprise Manager*, the reader should refer to Chapter V.

C.   JDBC™ CLASSES

JDBC™ implementation is provided by the *java.sql* package, which contains the classes and methods as shown in Table 2.1. In this section, we will attempt to give you a very broad overview of the JDBC™

9

object model, specifically looking at the primary purpose of each object. If additional information is required, the reader can find API documentation at [http://www.javasoft.com].

Object	Related Classes
Driver	*java.sql.Driver* *java.sql.DriverManager* *java.sql.DriverPropertyInfo*
Connection	*java.sql.Connection*
Statement	*java.sql.Statement* *java.sql.PreparedStatement* *java.sql.CalllableStatement*
ResultSet	*java.sql.ResultSet*
Errors/Warnings	*java.sql.SQLException* *java.sql.SQLWarning*
Metadata	*java.sql.DatabaseMetaData* *java.sql.ResultSetMetaData*
Date/Time	*java.sqlDate Java.sql.Time* *java.sql.TimeStamp*
Miscellaneous	*java.sql.Types* *java.sqlDataTruncation*

Table 2.1. JDBC™ Classes.

The topmost class in the JDBC hierarchy is the *DriverManager*. This class provides an indispensable object that you must explicitly create at least once in your JDBC™ program. The *DriverManager* is an important, necessary part of the object model because it maintains a list of registered database drivers, i.e. drivers that you intend to use in your application. When you create this object, its constructor requires that you pass it a string. This string represents the directory hierarchy and driver class file name on the remote machine. Without this information, there is no way the applet's *ClassLoader* could get the required files, and hence, communication with a remote data source would be impossible.

Once the driver is properly loaded, you need some means of indicating what data source you want to talk to, and where that data source resides. JDBC identifies a specific database by using a Uniform Resource Locator (URL) that has the following syntax:

jdbc:<subprotocol>:<subname:port/DSN>

By using the *DriverManager's getConnection* method, we provide this information. The URL that is passed to this method is:

jdbc:ff-microsoft://spitfire.avsafety.nps.navy.mil:1433/<DSN>

10

where DSN represents the Data Source Name of the database we want to connect to. The URL we use has the subprotocol *ff-microsoft*, which serves as a reference to the *FastFoward* driver we purchased. The subname indicates that our JDBC™ calls should go through the driver to port 1433 of the machine located at spitfire.avsafety.nps.navy.mil. Port 1433 is running Microsoft's BackOffice SQL Server 6.5 application. This server listens on port 1433 for incoming database requests, and if the remote user is authorized to issue commands, will process them and return the results via the driver that originally initiated communication. In this manner, the driver serves as a middleware solution for communicating with a DBMS server.

Should the process of loading the *DriverManager* succeed, then a *Connection* object is returned. It is from this object that we create *Statement* objects, which serve as the interface for executing SQL commands. By associating a *Statement* object with *Connection*, we know what data source we are interacting with at all times. There are different types of *Statement* objects, which we can create based on our needs. For example, we might use a regular *Statement* object to perform a one-time query, but should we find ourselves repeating the query with different criteria, we might opt to use a *PreparedStatement* which allows us to use the same query syntax and only change the criteria. If we have a procedure that is used repeatedly and consists of a sequence of SQL statements, we can use a *CallableStatement* to create, invoke, and pass parameters to a stored procedure.

The *ResultSet* represents a collection of rows that are the output of a SQL query. If no such rows exist, then the *ResultSet* object is consequently null. We use this object to navigate through the records, or tuples, that are returned from a query, and this object is related to a *Statement* because it is the latter object that executes a SQL query.

Java's™ strong typing and error handling requires that our interacting with a database contain a number of *try/catch* blocks to look for errors. If a problem does occur, a *SQLException* or *SQLWarning* object are created which we can then examine to determine what fault actually occurred. If necessary, we can also provide feedback to the client, or we can allow program execution to continue and take no action at all.

Later in this chapter we will demonstrate what happens when the user enters information which is already stored in the database.

JDBC™ also allows us to talk to a database that we know nothing about. We do this through the *MetaData* object which provides methods for examining a database's schema, or design. By allowing us to dynamically discover the contents of a database, we can create applications that do not have to change if the database does. The ACSA project did not use this object, and hence, if the databse is changed, the code will unfortunately have to be modified as well.

D.   SELECTING A SUITABLE JDBC™ DRIVER

The Sun Microsystems web site at [http://www.javasoft.com/products/jdbc/jdbc.drivers.html] offers an excellent, up-to-date discussion and listing of all available JDBC™ drivers. The database industry has categorized drivers into four categories as shown in Table 2.2:

DRIVER CATEGORY	DRIVER DESCRIPTION	ALL JAVA™
1	JDBC™-ODBC bridge plus ODBC driver	No
2	Native-API partly-Java™ driver	No
3	JDBC™-Net pure Java™ driver	Yes
4	Native-protocol pure Java™ driver	Yes

Table 2.2. Driver Categories [Ref. 3].

According to Sun, driver categories 1 and 2 are interim database connection solutions where pure Java™ drivers are not yet available. Their deficiencies stem from the fact that they are not written exclusively in the Java™ programming language, which means that they are not strongly typed, may fail to handle exceptions, and are not subject to the limitations imposed by the Java™ security model. Additionally, some of these drivers are not thread-safe and require binary code to be loaded on each client machine, which is a severe limitation the greater the audience size.

Since the ACSA prototype solution was a three-tier, client/server application, we limited our consideration to category 3 and 4 drivers. After reviewing the driver overview and product descriptions provided by the Javasoft web site, we selected a category 3 driver from IDSSoftware [http://www.idssoftware.com/jdriver.htm] and a category 4 driver from WebLogic, Inc. [http://www.weblogic.com/products/jdbckonaindex.html] to work with.

Both drivers provided a limited number of connections and had an evaluation period before their licenses expired. After experimenting with each driver and reviewing the applicable supporting documentation, we opted to use WebLogic's *FastForward* JDBC™ driver for three fundamental reasons:

1. The WebLogic driver had much better documentation and was easier to install and test. Included were numerous code examples that illustrated how to use the driver in both Java™ applications and applets.

2. The driver offered 15 simultaneous database connections, whereas the product from IDSSoftware only allowed ten. Additionally, the latter driver was almost identical in cost to the former, and used a proprietary server to communicate with data sources. Because SQL Server was installed and properly working on the prototype system, we wanted a driver that would interface with this server rather than a proprietary one whose reliability we knew nothing about.

3. WebLogic provided excellent support when development issues arose, and when we requested for an extended license, they were willing to give us six additional months for testing and evaluation with their product.

E.   WEBLOGIC'S FASTFORWARD DRIVER

Contact with WebLogic was either via the Internet or over the phone. Should future project enhancements be made and developers require knowledge of the driver, questions can be directed to Anil Bairwal [anil@mail.weblogic.com] via email or to other technical representatives by calling 415-659-2600.

Throughout two months testing and evaluation of the prototype system, there was only one observed problem that may or may not be related to use of the driver. After a brief revision to the source code for the *Results* applet, the application raised a *NullPointerException* and indicated that the maximum number of connections the driver could handle were in use. Knowing that this was not the case, we chose to shut down and reboot the server, which fixed the problem. Apart from having to do this on one other occasion, the incident has not been repeated.

For more information on the Weblogic driver, as well as the license agreement, see Appendix B. Driver documentation is on the prototype system and can be found in the *Weblogic* or *Connect* subdirectories on the D drive. At the time of this writing, WebLogic

was in the process of being acquired by BEA Systems. BEA has pledged to fully support and maintain all WebLogic products, and can be reached at [http://www.beasys.com/weblogic/].

F. APPLETS AND JDBC™

The ACSA project uses JDBC™ in Java™ applets for communicating with databases on a remote server. The source code fragment in Figure 2.2 illustrates how both applets open and then close a connection to the *Survey2_DB* database:

```
1: import java.sql.*;
<code omitted>
2: java.sql.Connection con;
3: java.sql.Statement stmt;
4: java.lang.String url = "jdbc:ff- microsoft://spitfire.avsafety.nps.navy.mil:1433/Survey2_DB";
<code omitted>
5: public void makeConnection(){
6:  try {
7:       Class.forName("connect.microsoft.MicrosoftDriver").newInstance();
8:       con = DriverManager.getConnection(url, "sa", "");
9:       stmt = con.createStatement();
10:  }
11: · catch (Exception e) {
12:       textDisplayArea.appendText(e.toString());
13:  }
14: }//end makeConnection()
<code omitted>
15: public void closeConnection() {
16:  try{
17:    con.close();
18:    stmt.close();
19:  }
20:  catch ( java.sql.SQLException e ){
21:    textDisplayArea.appendText( e.toString() );
22:  }
23: }//end closeConnection
```

Figure 2.2. Making a connection to the *Survey2_DB.*

Some explanation of what this code does is necessary, so we will briefly highlight what is occuring here. You will notice that the source code that is not relevant to this discussion has been purposely omitted. For a complete copy of applet code, the reader should refer to Appendix C for the *Results* applet and Appendix D for the *Survey* applet.

In line 1, the *import java.sql.** statement allows us to use the classes and interfaces that are defined in the JDBC™ API. Lines 2 and 3 declare a *Connection* and *Statement* object respectively, and line 4

initializes the string literal *url* to the location where the *Survey2_DB* is located.

The method *makeConnection* that follows is available for the applet to call any time it wants to make a connection to the *Survey2_DB*. It is on line 7 where this method signals the browser's *ClassLoader* that the applet will need some additional files. By indicating the directory hierarchy and the class name that is required, the browser can go back to the web server the applet was downloaded from and retrieve those files. In this case, we need the WebLogic driver that communicates with the SQL Server, which is in a file appropriately named *MicrosoftDriver*. This file is located two subdirectories below the directory that contains the applet source code.

The *newInstance* method creates a new instance of this *MicrosoftDriver* class object that we can use. During this process, it is registered with the *DriverManager*. While this occurs, the applet's main thread of execution is paused so that downloading can complete and subsequent calls that might need the driver will not raise exceptions. If there are no problems loading the necessary files, execution continues with line 8, where we make a call to the overloaded *DriverManager* function *getConnection*. The *DriverManager* constructs a *Connection* object and returns a reference to it, which we then assign to the variable *con*.

By keeping a reference to this object, we can then use it to create *Statement* objects, as is done in line 9. Through the association of the two objects, any *Statement* that is executed knows to which connection it belongs and will only affect that data source. If a query needs to occur on multiple data sources, separate *Connection* and *Statement* objects must be created and maintained.

The *getConnection* function we use requires a username and password, so we pass the string *sa* (System Administrator) as the client's identify, and a null password. This information is used by SQL Server to authenticate the client that is trying to use the *Survey2_DB*. SQL Server allows you the ability to restrict access to databases on your machine by using Windows NT user accounts, but the default permission is to allow everyone access. Because the prototype system does not have the Guest account enabled, only users who have properly responded to Windows NT's challenge/response authentication mechanism will be

allowed to use these applets. Additional discussion on this subject is included in Chapter V. For now, you should be aware that we can change these user names and passwords as we desire, and only a single line of code will have to be modified.

Once a connection to the database is established, it is kept open until it is no longer needed. At that time the program calls the method *closeConnection*, which simply frees the resources held by the *Connection* and *Statement* objects. The next section illustrates the use of JDBC™ to check a client's survey number and determine its validity.

G.   CHECKING SURVEY NUMBERS

Although the ACSA program is highly automated, some administration will be required at the local level. Because we wanted to protect the integrity of the data, it is not enough that users simply enter a username and password to access the web site. If this were the case, then it is possible and very likely that one user could enter multiple surveys, which would skew the data and adversely affect any analysis. So, before a squadron member can blindly submit a survey, we decided to to levy a requirement that the individual be issued a survey number.

The local survey administrator, the Aviation Safety Officer (ASO) or other designated representative, will have to receive these numbers from the ASCA site administrator before the site can be accessed and used. These numbers are found in the *Num_List_DB* database, which was constructed as follows:

Field	Description
Unit_ID	Database primary key; text representing the individual squadron, e.g. "HM10", "VF101"
Start_Number	Lower limit of authorized survey numbers for this particular squadron.
End_Number	Upper bound on authorized survey numbers for this particular squadron.

Table 2.3. The *Num_List_DB* schema.

The method of operation is very much the same if you are using the HTML or applet interface. At the time that the client is ready to submit his data, he is queried for a squadron and survey number. After entering the information, the client's information is either accepted as valid and he is redirected to the page *ThankYou.html*, or the squadron or survey number are invalid and the user is redirected to the page *NumberError.html* which notifies him of the problem.

The code that validates the survey number is handled by the method *checkSurveyNumber*. This function uses JDBC™ to connect to the *Num_List_DB* database in the same fashion that connection was made to the *Survey2_DB* database described earlier. The connection initiating source code is for *checkSurveyNumber* is located on lines 15, 21 and 22 of Figure 2.3. A valid survey number can include the *Start_Number,* *End_Number,* or any number between them.

```
1:  // Function:    checkSurveyNumber()
2:  // Parameters:  unit - the unit submitting the survey
3:  //              squad - the unit's squadron number
4:  //              surv - the unit's survey number
5:  //Return Type: boolean - true if the survey number is valid for the given unit and squadron, false
6:  //              otherwise
7:  //Purpose:      Confirms if the survey number is valid or not.  If it is valid, then the client can submit
8:  //              the survey, otherwise, the client is informed that the survey number is invalid.
9:  //-------------------------------------------------------------------------------------------------------
10: private boolean checkSurveyNumber(java.lang.String unit,
11: java.lang.String squad, java.lang.String surv){
12: java.lang.String communities[] = {"HC", "HCS", "HM", "HMH", "HMT", "HS", "HSC", "HSL",
13:                                    "VAQ", "VAW", "VF", "VFA", "VMAQ", "VMFA",
14:                                    "VP", "VQ", "VR", "VRC", "VS", "VT", "VX", "Other"};
15: java.lang.String url = "jdbc:ff-microsoft://spitfire.avsafety.nps.navy.mil:1433/Num_List_DB";
16: int start = -1, stop = -1;
17: boolean valid = false;
18:     if (communityChoice.getSelectedIndex() != -1){
19:       unit = new String(communities[communityChoice.getSelectedIndex()] + squad);
20:       try {
21:         Class.forName("connect.microsoft.MicrosoftDriver").newInstance();
22:         java.sql.Connection con = DriverManager.getConnection(url, "sa", "");
23:         java.sql.PreparedStatement stmt = con.prepareStatement("SELECT * FROM Num_List
24:                                                 WHERE  Unit_ID = ?");
25:         stmt.setString(1, unit);
26:         java.sql.ResultSet rs = stmt.executeQuery();
27:         if (rs != null){
28:           while (rs.next()){
29:             rs.getString(1);
30:             start = rs.getInt(2);
31:             stop = rs.getInt(3);
32:           }
33:         }
34:         if ((Integer.parseInt(surv) >= start) && (Integer.parseInt(surv) <=stop)){
35:           valid = true;
36:           responseText.appendText("Your survey number is valid... Please wait...\n");
           }
37:         else responseText.appendText("You entered an INVALID number!!!\n");
38:         stmt.close();
39:         con.close();
40:       }
41:       catch (Exception e) {
42:         responseText.appendText(e.toString() + "\n");
43:       }
44:     }
```

17

```
45:    return valid;
46: }//end checkSurveyNumber()
```

Figure 2.3. Validating the client's survey number.

So what is occuring here? If you look at lines 16 and 17, you see that we initialize the variables *start,* and *stop* to -1 and *valid* to false. Line 19 looks at a *Listbox* component on the *Survey* applet to determine if the user has selected a community. If so, the value he selected is concatenated with his squadron number and put into the String object *unit.* With that done, we are ready to connect to the database.

Lines 21 and 22 open a connection to the *Num_List_DB* database in the same way a connection was established to the *Survey2_DB.* We used a *PreparedStatement* object to form the query we are going to perform because we know that each time we perform this query, the only thing that changes is the value for *unit.* Using a *PreparedStatement* improves program performance because these queries do not have to be recompiled every time they are submitted to the database. The benefit from this feature however is limited in **checkSurveyNumber** because the SQL statement is executed only once and a connection to the database is not maintained.

In line 25, the *stmt.setString(1, unit)* looks for the first "?" of our *PrepardeStatement* object *stmt,* and it replaces that symbol with the value held by the string *unit.* With that substitution made, the SQL statement is complete and ready to be passed to the database. In this case, our SQL query

*SELECT * FROM Num_List WHERE Unit_ID = unit*

selects all entries from the table *Num_List* in the database *Num_List_DB* where *Unit_ID* equals the value of *unit.* The results from this query are then executed and go into the *ResultSet* object *rs* on line 26. *rs* is assigned the tuples of any returned records and this object can then be used to navigate the returned information.

It is important to know that JDBC™ uses a forward-only cursor to reference the current tuple in a *ResultSet* object. This cursor is initially positioned before the first row of any returned records, and once we move ahead, there is no way we can look at a previous record unless we perform the original query again. Because the cursor is positioned prior to the first record, you must make a call to the

ResultSet's next method, which moves the cursor to the first record if one exists. Subsequent calls to this method continue to advance the cursor.

Knowing how to navigate allows you to move from one record to another, but in order to retrieve information from the database, you need to use a *getXXX* method (where *XXX* is replaced by the data type of the field you are interested in examining).

If our *ResultSet* does indeed have a record to look at (our query is guaranteed to return either one record or none because multiple units cannot exist in our database), we are then positioned at the first record by line 28. Line 29 uses the *getString(1)* method to indicate that the *ResultSet* should return a string containing the value of the first field. Because we do not assign this value to any variable, it is effectively discarded and does not serve any purpose.

Lines 30 and 31 use similar method calls to determine what the legitimate *start* and *stop* numbers are for the client's *unit*. If the unit was found, then we do some comparison in lines 34 through 37 to see if the client entered valid information. If he did, then processing continues, otherwise he is alerted that his number is not valid.
At the conclusion of this process, the resources the applet uses are free and the connection is severed so that other users may submit information. By only opening the connection when you need it, and closing it shortly thereafter, you can avoid possible problems posed by the license limitations of the driver.

All possible conditions and their outcome are covered by this application and enumerated in Table 2.4 below:

SQUADRON	ID	OUTCOME
Valid squadron	Valid survey number	Data submitted.
Valid squadron	Invalid survey number	Error returned.
Invalid squadron	Valid survey number	Error returned.
Invalid squadron	Invalid survey number	Error returned.

Table 2.4. Extending the possibilities of client input.

H.   MAKING DATABASE ENTRIES

The function *makeDatabaseEntry* (see Figure 2.4) is used for inserting data into the *Survey2_DB* database. Since a database *UPDATE* operation changes the contents of the database, we need some way of ensuring that all of the information the client submits is entered.

This is a concern because any number of undesirable events can occur, e.g. power loss, loss of connection, etc…, which can preclude this from happening. In order to solve this problem, we use what is known as a "transaction". JDBC™ ensures database changes are made correctly with a "transaction". A transaction consists of a sequence of SQL calls that constitute an atomic unit of work. Transactions provide ACID properties: atomicity, consistency, integrity of data, and durability of database changes [Ref. 2].

By default, JDBC™ treats every sequential line of code as a transaction. In order to group multiple lines of code into one transaction we must disable this setting. In line 39, we call the *Connection* object's *setAutoCommit* method and pass it the value *false* to indicate that data should not be committed until we have everything we need.

This step disables the auto-commit mode and treats all the statements between lines 39 and 63 as one atomic statement. All JDBC™ calls are then either committed to the database, or the transaction is rolled back, or aborted, and data is restored to its original value.

```
1.  //Function:    makeDatabaseEntry()
2.  //Parameters:  squadN - the unit's squadron number
3.  //             surveyN - the client's survey number
4.  //Return       Type: None
5.  //Purpose:     The client's data has already been checked, and the survey number has been
6.  //             been validated prior to this function being called.  Once this procedure is
7.  //             completed, the client's data has been inserted into the SQL 6.5 database.
8.  //             Entries are made in both the biographical and questions table, and if either
9.  //             insertion operation fails, then the transaction is rolled back (i.e. it is
10. //             removed from the database).
11. //--------------------------------------------------------------------------------------
12.    private void makeDatabaseEntry(java.lang.String squadN, java.lang.String surveyN){

13.       java.lang.String url = "jdbc:ff-microsoft://spitfire.avsafety.nps.navy.mil:1433/Survey2_DB";
14.       java.lang.String designator[] = {"Pilot", "NFO", "Other"};
15.       java.lang.String communities[] = {"HC", "HCS", "HM", "HMH", "HMT", "HS", "HSC", "HSL",
16.                                          "VAQ", "VAW", "VF", "VFA", "VMAQ", "VMFA",
17.                                          "VP", "VQ", "VR", "VRC", "VS", "VT", "VX", "Other"};
18.       java.lang.String service[] = {"USN", "USMC", "Other"};
19.       java.lang.String DH, ST, LOC;
20.       //independently determine which checkboxes are selected
21.       if (DepartmentHead.getSelectedCheckbox() == yesRadioButton)
22.          DH = new String("Yes");
23.       else DH = new String("No");
24.       if (status.getSelectedCheckbox() == regularRadioButton)
25.          ST = new String("Regular");
26.       else if (status.getSelectedCheckbox() == activeRadioButton)
27.          ST = new String("Active Reserve");
28.       else ST = new String("Drilling Reserve");
```

```
29.     if (location.getSelectedCheckbox() == eastButton)
30.        LOC = new String("East Coast");
31.     else if (location.getSelectedCheckbox() == westButton)
32.        LOC = new String("West Coast");
33.     else LOC = new String("Other");
34.     java.sql.Connection con = null;
35.     try {
36.        Class.forName("connect.microsoft.MicrosoftDriver").newInstance();
37.        Driver driver2 = new connect.microsoft.MicrosoftDriver();
38.        con = DriverManager.getConnection(url, "sa", "");
39.        con.setAutoCommit(false);
40.        java.sql.Statement stmt1 = con.createStatement();
41.        java.sql.Statement stmt2 = con.createStatement();
42:        stmt1.executeUpdate("INSERT INTO Biographical_Data VALUES('" +
43:                  communities[communityChoice.getSelectedIndex()] + "'," +
44:                  Integer.parseInt(squadN) + "," +
45:                  Integer.parseInt(surveyN) +"," +
46:                  rankChoice.getSelectedIndex() + ",'" +
47:                  designator[designationChoice.getSelectedIndex()] +"'," +
48:                  Integer.parseInt(flightHoursText.getText()) + "," +
49:                  Integer.parseInt(typeHoursText.getText()) + ",'" +
50:                  DH + "','" +
51:                  ST + "','" +
52:                  service[serviceChoice.getSelectedIndex()] + "','" +
53:                  LOC + "')");
54:        java.lang.String operation = "INSERT INTO Questions VALUES('" +
55:                  communities[communityChoice.getSelectedIndex()] + "'," +
56:                  Integer.parseInt(squadN) + "," +
57:                  Integer.parseInt(surveyN) +",";
58:        for (int ix=0; ix<NUM_QUESTIONS-1; ix++)
59:           operation += (choices[ix].getSelectedIndex() + ",");
60:        operation += choices[NUM_QUESTIONS-1].getSelectedIndex() + ")";
61:        stmt2.executeUpdate(operation);
62:        con.commit();
63:        con.setAutoCommit(true);
64:        stmt1.close();
65:        stmt2.close();
66:        con.close();
67:        responseText.appendText("Survey successfully submitted. Thank you for your input.\n");
68:        miNew_ActionPerformed(null);
69:     }
70:     catch (Exception e){
71:        responseText.appendText("Survey not submitted!  Processing error encountered:\n");
72:        responseText.appendText(e.toString() + "\n");
73:        try {
74:           if (con != null)
75:              con.rollback();
76:        }
77:        catch (SQLException ex){
78:           responseText.appendText("Survey not submitted!  Processing error encountered:\n");
79:           responseText.appendText(ex.toString() + "\n");
80:        }
81:     }
82: }//end makeDatabaseEntry()
```

Figure 2.4. The function *makeDatabaseEntry*.

The *Survey2_DB* database contains the two tables *Biographical_Data* and *Questions* linked with a composite key of consisting of three fields: *Community, Squadron_Number* and *Survey_Number.* By the time the function *makeDatabaseEntry* is called, the client's data has already been checked for correctness and the survey number has been validated.

This method receives as parameters the client's unit and survey number. Lines 35 through 41 operate as before, where we load the database driver and wait for the files to be downloaded to the client's browser. Once this is done, we are ready to take the biographical information from the applet and prepare it for insertion into the database. This is done by the 12 lines from 42 to 53. A separate *Statement* object from the one we will use to input the question responses is used because none of the code is actually being executed. Rather, it is simply being queued for execution and will occur all at once because it is part of a transaction. If we used the same *Statement* object but changed its value, we would wind up attempting to only insert the responses to the 61 questions.

Lines 42 through 52 use Java™ GUI object methods to retrieve the selections the client has made. Because these GUI components are drop-down or Choice containers, we know what their values will be based on the index number that is selected. For this reason, we can appropriately create arrays that contain the values of the survey, and then index into these arrays to grab the values the client has selected.

After processing of the biographical information is complete, we move on to lines 54 through 60 which append to the string *operation* the numeric values of the questions. When we reach line 62, we execute both *Statement* objects and commit the data to the SQL 6.5 database. If a problem is encountered, exception handling moves the point of execution to line 71 where three things are done: the client is informed that a problem was encountered, information as to what occurred is detailed, and if a connection is still active, the data is rolled back to its prior state. This means that if the record did not exist before we attempted to insert it, it will not exist afterwards.

I. THE RESULTS APPLET

Clients have the ability to graphically view database results through the *Results* applet shown in Figure 2.5. In this section, we will review using this applet and the role of JDBC™ in accessing data from the database.

Command Safety Assessment

Number of surveys: 35

Question 1: My command conducts adequate reviews and updates of safety standards and operating procedures.
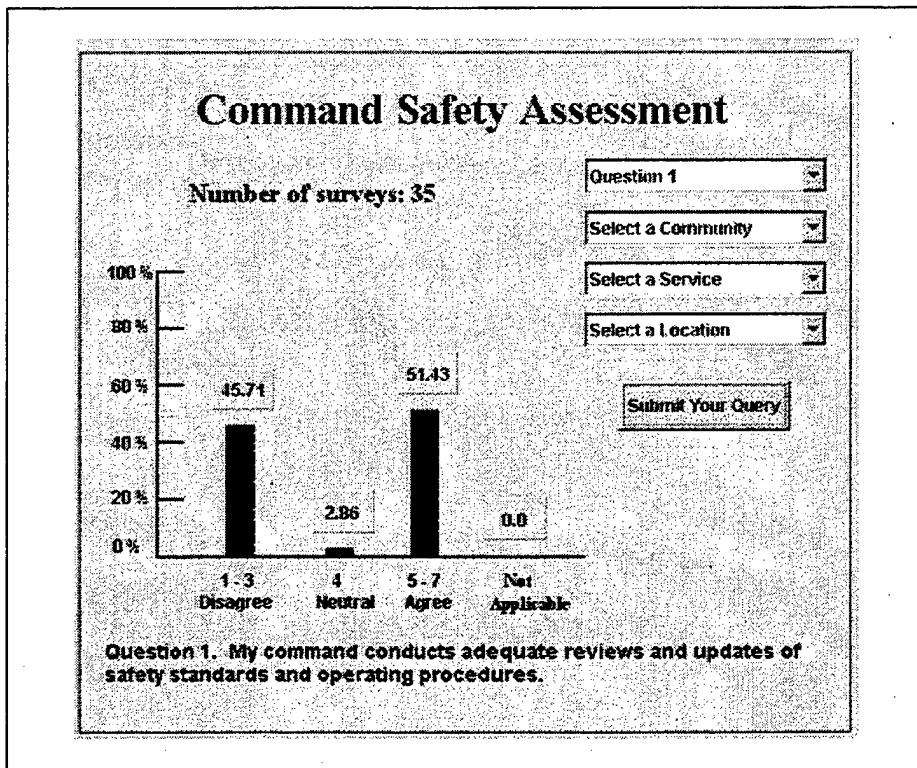
Figure 2.5. The *Results* applet.

This program displays results for individual survey questions based upon a client-selected matrix of community, service and location options. Selecting one of the 61 survey questions forms the root of every SQL query statement passed to the database. All database queries must be formed using this question as the selection basis, with results being displayed one question at a time. For example, in Figure 2.5, a client selected question 1 as the root of their query.

An overview of the SQL creation process is presented in Figure 2.6. Every query begins with a root which is initialized in the method *processQuery*. Depending on the client's selections, one of eight functions is then called to complete constructing the specific SQL statement according to the criteria the client wishes to examine.

Finally, the function *countResponses* is called to execute the query and
tabulate the results that need to be returned.
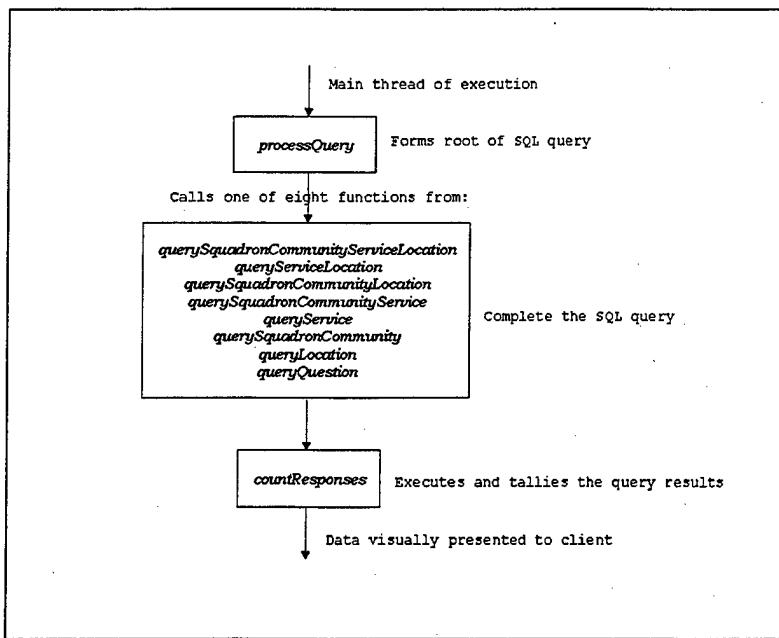
Figure 2.6. Building a query.

Queries are initiated by the function *processQuery*. The source code
for *processQuery* is displayed in Figure 2.7. The root of every query
begins on line 12 where we initialize the string object *query* to the
string literal "**SELECT Q**". The function then looks at each drop-down
box, determining according to the client's selections which call it
next needs to make. The decision of which call to make next is made by
the series of **if** and **else if** clauses shown from lines 14 through 45.

An hypothetical example will best serve as a good illustrative
guide as to what occurs on these lines. Let's suppose we have a client
who is interested in viewing the results of question one for all U.S.
Naval aviation commands that are located on the East Coast. He makes
his selections accordingly:

GUI component	Set to	Meaning
Select a Question	Question 1	Query question number 1.
Select a Community	Default value	Query should include responses from all communities.
Select a Service	USN	Responses should be tallied only if the service of the individual submitting the survey is USN.
Select a Location	West Coast	Squadron should be on the West Coast.

Table 2.5. Hypothetical selection box entries.

Using this example, we start evaluating the client's selections on line 14. The method *getSelectedIndex* belongs to a Java™ *Choice* object and returns the index number of the client's selection (no selection returns a value of -1). Because our default selections are strings of text instructing the user what to do, a value of 0 is returned if the client does not explicitly choose a query criteria. In the example above, the only time we will get a 0 is when we use the *getSelectedIndex* method of the GUI component that prompts the user for a community. This component, which we call the *squadronCommunityBox*, causes the **if** condition starting on line 14 to fail, and execution continues on line 20 where all the components we check have client selections. Thus, we pass the parameter *query* to the method *queryServiceLocation*, which will append the proper SQL syntax to the end of the string literal.

```
1:  //Function:      processQuery()
2:  //Parameters:    None
3:  //Return Type:   None
4:  //Purpose:       Determines the specific query to be passed to the database
5:  //               from the user's selections of: question number, community,
6:  //               service and or location.  Each query selection calls
7:  //               the appropriate query function, e.g. queryService(query).
8:  //               Calls clearDisplay() to clear the display in the event of
9:  //               multiple user queries.
10: //-----------------------------------------------------------------------------
11: public void processQuery() {
12:    java.lang.String query = "SELECT Q" ;
13:    clearDisplay();

14:    if (  ((serviceChoiceBox.getSelectedIndex() !=0) && (serviceChoiceBox.getSelectedIndex() != -1))
15:        && ((squadronCommunityBox.getSelectedIndex() !=0) &&
16:            (squadronCommunityBox.getSelectedIndex() != -1))
17:        && ((squadronLocationBox.getSelectedIndex() !=0) &&
18:            (squadronLocationBox.getSelectedIndex() != -1)) )
19:        querySquadronCommunityServiceLocation(query);

20:    else  if ( ((serviceChoiceBox.getSelectedIndex() !=0) && (serviceChoiceBox.getSelectedIndex()
21:               != -1))
22:            && ((squadronLocationBox.getSelectedIndex() !=0) &&
23:                (squadronLocationBox.getSelectedIndex() != -1)) )
24:            queryServiceLocation(query);
*
25:    else  if ( ((squadronCommunityBox.getSelectedIndex() !=0) &&
26:               (squadronCommunityBox.getSelectedIndex() != -1))
27:            && ((squadronLocationBox.getSelectedIndex() !=0) &&
28:                (squadronLocationBox.getSelectedIndex() != -1)) )
29:            querySquadronCommunityLocation(query);

30:    else if ( ((serviceChoiceBox.getSelectedIndex() !=0) && (serviceChoiceBox.getSelectedIndex()
31:               != -1))
32:            && ((squadronCommunityBox.getSelectedIndex() !=0) &&
```

```
33:                (squadronCommunityBox.getSelectedIndex() != -1)) )
34:            querySquadronCommunityService( query );

35:        else if ( (serviceChoiceBox.getSelectedIndex() !=0) &&
36:                (serviceChoiceBox.getSelectedIndex() != -1) )
37:            queryService(query);

38:        else if ( (squadronCommunityBox.getSelectedIndex() !=0) &&
39:                (squadronCommunityBox.getSelectedIndex() != -1) )
40:            querySquadronCommunity(query);

41:        else if ( (squadronLocationBox.getSelectedIndex() !=0) &&
42:                (squadronLocationBox.getSelectedIndex() != -1) )
43:            queryLocation(query);
44:        else
45:            queryQuestion(query);
46: }//end processQuery()
```

Figure 2.7. The function *processQuery*.

QueryServiceLocation builds the specific SQL statement from the client's question, service, and location selections. In lines 14 through 19 of Figure 2.7, we begin the concatenation process by adding the required query composite key fields. Next, we add on the value for the client's service and location selections. This completes the SQL query statement, and it is sent to *countResponses* for futher processing.

```
1:  //Function:     queryServiceLocation()
2:  //Parameters:   java.lang.String query, the root query passed from processQuery
3:  //Return Type:  None
4:  //Purpose:      Receives the initial query from processQuery()and
5:  //              concatenates the remaining SQL syntax to the string literal.
6:  //              This specific query calculates the question number,
7:  //              service, location tuple.  Calls countResponses()
8:  //              to calculate the number of tuples for this query.
9:  //-----------------------------------------------------------------------------------
10: public void queryServiceLocation( java.lang.String query ){
11:   textDisplayArea.append("\nQuery -> queryServiceLocation ") ;
12:   textDisplayArea.append("\nService ->" + service[serviceChoiceBox.getSelectedIndex() -1] ) ;
13:   textDisplayArea.append("\nLocation ->" + location[squadronLocationBox.getSelectedIndex() -1] ) ;
14:   query += questionBox.getSelectedIndex() + " FROM Biographical_Data INNER JOIN Questions" +
15:     "ON Questions.Community = Biographical_Data.Community AND " +
16:     "Questions.Squadron_Number = Biographical_Data.Squadron_Number AND " +
17:     "Questions.Survey_Number = Biographical_Data.Survey_Number WHERE " +
18:     "Biographical_Data.Service = '" + service[serviceChoiceBox.getSelectedIndex() -1] + "' " +
19:     "AND Biographical_Data.Location = '" + location[squadronLocationBox.getSelectedIndex() -1]+
20:     "";
21:   textDisplayArea.append("") ;
22:   textDisplayArea.append("\nquery ->" + query ) ;
23:   textDisplayArea.append("") ;
24:   countResponses( query );
25: }//end queryServiceLocation()
```

Figure 2.8. The function *queryServiceLocation*.

The function *countResponses* processes the *SELECT* query statement by calling the method *executeQuery*. Any data that matches the criteria the user seeks is returned to the *ResultSet* object *rs*. We then step through each record in lines 19 through 30 and tabulate results based on a range of responses. Once this is done, the method *displayData* (see Figure 2.10) is called to create a histogram much like the one shown earlier in this section.

```
1:  //Function:     countResponses()
2:  //Parameters:   java.lang.String query, the concatenated SQL query to be passed to the ASCA database.
3:  //Return Type:  None
4:  //Purpose:      Receives and executes the specific database query. Counts
5:  //              each individual result tuple recording the number of
6:  //              records in each category.  Counts the total number
7:  //              of records received in numTotal.
8:  //-----------------------------------------------------------------------------------------------
9:  public void countResponses( java.lang.String query ){

10:   try{
11:     ResultSet rs = stmt.executeQuery( query );

12:     //initialize variables for multiple queries
13:     numTotal =0;
14:     numDisagree =0;
15:     numNeutral =0;
16:     numAgree =0;
17:     numNotApplicable =0;
18:     rowCount =0;

19:     while (rs.next()) {
20:       int x = rs.getInt(1);
21:       if ((x >= 0 ) && ( x <= 2 ))
22:         numDisagree++;
23:       else if (x == 3)
24:         numNeutral++;
25:       else if (x > 3 )
26:         numAgree++;
27:       else if (x == -1)
28:         numNotApplicable++;
29:       rowCount++;
30:     }//end while

31:     numTotal = (numDisagree + numNeutral + numAgree + numNotApplicable);
32:     rs.close();
33:   }
34:   catch (Exception e) {
35:     textDisplayArea.appendText(e.toString());
36:   }
37: }//countResponses()
```

Figure 2.9. The function *countResponses*.

```
 1:  //Function:       displayData()
 2:  //Parameters:     None
 3:  //Return Type:    None
 4:  //Purpose:        Checks the total number of records for this display,
 5:  //                numTotal, against the minimum number of records required
 6:  //                to display query data, MIN_SURVEY_NUM. This feature protects
 7:  //                individual anonymity at the beginning of the survey
 8:  //                process.  If the query returns more than MIN_SURVEY_NUM
 9:  //                records then this function sets visible each display object.
10: //------------------------------------------------------------------------------------------------------
11:  public void displayData(){

12:      //check that the required number of records exist to display data
13:      if (numTotal < MIN_SURVEY_NUM) {
14:        clearDisplay();
15:        //required number did not exist, display no results alert
16:        noResultsLabel.setVisible(true);
17:        processingQueryLabel.setVisible(false);
18:        return;
19:      }

20:      //required number of records exists, displaying results
21:      else {
22:        clearDisplay();
23:        try {
24:          //displays the corresponding survey question
25:          questionDisplayLabel.setText("Question " + questions[questionBox.getSelectedIndex()-1]);
26:          questionDisplayLabel.setVisible(true);

27:          //display the number of surveys used to calculate the display column data
28:          numSurveysLabel.setText("Number of surveys: " + rowCount);
29:        } catch(java.beans.PropertyVetoException e) {}
30:        processingQueryLabel.setVisible(false);
31:        numSurveysLabel.setVisible(true);
32:        noResultsLabel.setVisible(false);
33:        disagreeColumn.setVisible(true);
34:        disagreeColumn.setBounds(100,Y_coordDisagree,COLUMN_WIDTH,(int)disagreePixelHeight);
35:        neutralColumn.setVisible(true);
36:        neutralColumn.setBounds(170,Y_coordNeutral,COLUMN_WIDTH,(int)neutralPixelHeight);
37:        agreeColumn.setVisible(true);
38:        agreeColumn.setBounds(230,Y_coordAgree,COLUMN_WIDTH,(int)agreePixelHeight);
39:        notAppColumn.setVisible(true);
40:        notAppColumn.setBounds(295,Y_coordNotApp,COLUMN_WIDTH,(int)notAppPixelHeight);
41:        setPercentLabelVisible();
42:      }
43: }//end displayData()
```

Figure 2.10. The function *displayData*.

J.   CONCLUDING REMARKS

JDBC™ is Java™'s ODBC solution. While JDBC™ is an easy object model to learn and use, you need to remember that it is only available

to Java™ applications and applets. If you are creating a program in another language, you will need to find an alternative means of interacting with data sources (such as ADO, which is discussed in Chapter IV).

Both ACSA applets use JDBC™ to talk to the remote databases we developed for this project. If the back-end database is ever changed, it will be necessary to alter the manner in which these applets parse information into retrieve information from the database. While nothing can be done to preclude this foreseeable event, the ability to manipulate survey questions, client choices, and presentation styles could be done if the design of the applets were rebuilt as reusable JavaBean components. Because the initial development of the prototype system had to be completed fairly quickly, this was not done, but the process of extending the code to allow greater system flexibility should not be difficult to do.

III. JAVA™ SERVLETS

A.   THE COMMON GATEWAY INTERFACE (CGI)

During the early, formative years of the World Wide Web (WWW) (circa 1990), the Common Gateway Interface (CGI) specification was developed. Prior to the CGI concept, web pages were static displays of information (primarily text and graphics) that were conveyed to clients through the use of the Hypertext Markup Language (HTML). There was relatively little interaction between a client and server, and the connection between the two was severed as soon as the server was done processing the client's request. With CGI, however, clients could now do other things - such as run programs on remote web servers. These programs could be initiated either directly, by referencing the name of the program in a Uniform Resource Locator (URL), or more commonly, through the use of an HTML object known as a form. Via these means, a CGI program could parse form input into a database, generate new HTML pages "on-the-fly" and return them to the client, or perform some other action altogether. The ability to produce dynamic content exponentially increased the amount of client-server interaction, made the web a more appealing and useful information medium, and ensured that CGI would come into widespread acceptance.

As CGI was used more frequently, developers began to take note of its vulnerabilities. Chief among a long list of concerns was security. CGI scripts and programs were only as good as the programmer that developed them. Even experts at times unknowingly released flawed CGI programs (such as phf, a CGI program that shipped with the distribution of the National Center for Supercomputing Application's (NCSA) web server and allowed hackers to gain remote access to files). The exploit of such programs has served to make web administrators more security conscious, but it has not drastically reduced the use of CGI.

The other notable problem with CGI was the manner in which client requests were processed. When the web server received a request to run a CGI program, the server would generate a list of environmental variables which were then passed to the CGI program once it was loaded into memory. Regardless of whether the program accessed these variables or not, the environment's state had to be sent to the program. If there were multiple, simultaneous requests for the same CGI program, each request had to be satisfied individually, i.e., the server had to

generate and pass *n* sets of environmental variables for *n* requests, and the program was loaded and executed in its own memory space *n* times. Hence, *n* requests resulted in *n* separate threads of execution. This behavior was considered acceptable when a CGI program was requested once per minute. However, sites that were viewed millions of times and made extensive use of CGI programs sometimes found that the large volume of client requests either crashed the server or brought it to its knees.

These problems eventually had fixes. A program called CGIWrapper reduced the risk of CGI exploits, and an alternative CGI model (called Fast CGI) was developed that lessened server overhead and improved system performance. As with anything else, however, the progression of time gave rise to new, considerably safer technologies that could accomplish the same tasks in far less time. The new buzzwords became Active X, Active Server Pages (ASP), Java™ applets, JavaBeans, etc. Still, CGI somehow became entrenched in web site development, and it never did die. For this reason, and to present via example how CGI works, we continue with a brief CGI demonstration.

B.   A CGI DEMONSTRATION

A diagram perhaps best illustrates the client/server connection. The simplistic example we present at this juncture is representative of what happens when a client uses a browser to request a web page:
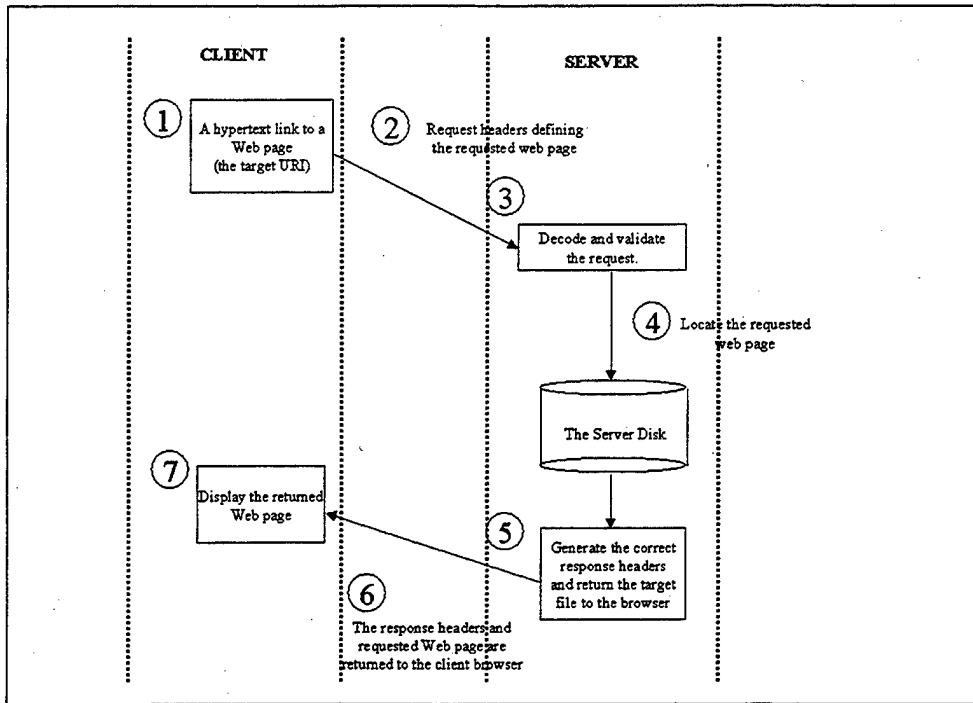
Figure 3.1. The Client/Server Connection [Ref. 4].

The seven-step process of serving a file is identical no matter what Multipurpose Internet Mail Extension (MIME) format is being served. An important distinction to make here, however, is that in the case of serving an HTML file, the web server automatically generates and sends the correct response headers for you. This is not the case with a CGI program, which must take care to explicitly do this. For example, if the programmer fails to return a correct response header, then the client is generally informed that the server encountered an internal error while it tried to process the CGI program.

Once the correct headers are sent, the process of generating HTML can begin. CGI programs can be written in any number of programming or scripting languages. Typically, developers use C or the Practical Extraction and Reporting Language (PERL). The short PERL script that follows was placed in the cgi-bin directory of our default web site running Microsoft's Internet Information Server (IIS 4.0). It is worth noting that IIS does not come with a PERL interpreter, and there are additional steps that need to be taken should you desire to support PERL scripts under IIS:

1. Download PERL from Hip Communications, Inc. [http://info.hip.com], and install it into its own directory independent of all web directories. Under no circumstances should the PERL executable (perl.exe) or its libraries be placed directly under the HTTPd software, specifically, under the cgi-bin directory [Ref. 5].

33

This placement exposes the executable to the outside world and, quite literally, any other machine on the Internet can point a script to your /cgi-bin/, overload your machine, and considerably reduce your available bandwidth [Ref. 5].

2. By using the Microsoft Management Console or the HTML Administrative interface to IIS, you can select the cgi-bin directory (or whatever location you choose to place your scripts in) from the default web site and modify its application mappings. Specifically, you should add the .pl or .cgi extension and map it to the PERL executable as demonstrated in the following IIS Management Console caption:
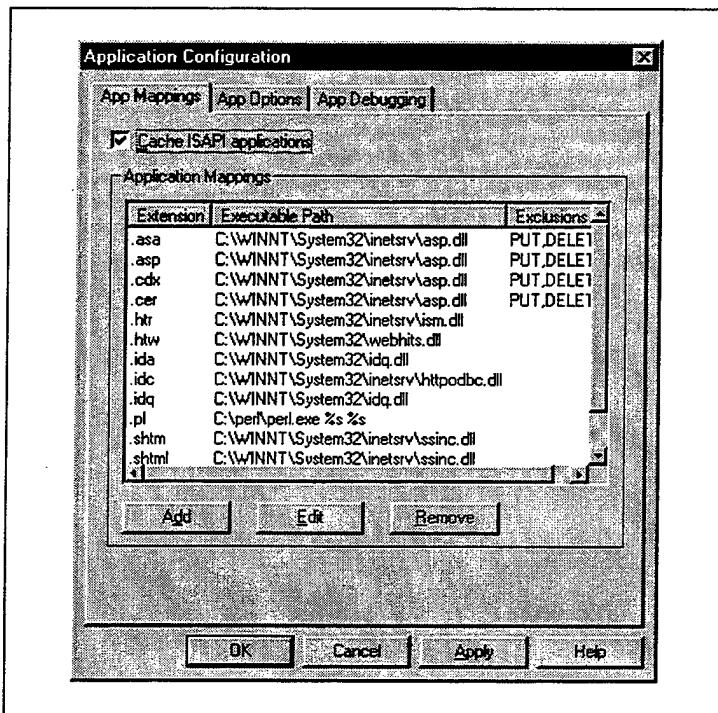
Figure 3.2. IIS Management Console (MMC) - using MMC to map the *.pl* extension to the PERL executable.

With the mapping complete (so that the web server now knows where the interpreter resides), and some PERL code written and in place, you can begin testing your script. You need to ensure that the mapping to the PERL executable has "%s %s" appended to it (The "%s %s" is case sensitive, so "%S %S" will not work [Ref. 6]). If this mapping isn't done correctly, CGI program execution will timeout under IIS after 900 seconds, and the server will fail to return a response.

The example PERL script we are going to use is only seven lines in length (see Figure 3.3). This scripts determines and tells the user what environmental variables the web server it is executed on supports (we can use this script on any web server that has and supports a PERL

interpreter, although configuring PERL will most likely differ from what has previously been described).

```
1: print "HTTP/1.0 200 OK\n";
2: print "Content-Type: text/html\n\n";
3: $i = 0;
4: while (($key, $val) = each %ENV) {
5:    print "i = ", $I, " ", "$key = $val<BR>\n";
6:    $i++;
7: }
```

Figure 3.3. The file *cgivars.pl*.

The first two lines of *cgivars.pl* are HTTP response headers. Line 1 is the HTTP Status header, which informs the client's browser what HTTP protocol version the program will communicate with (in this case, version 1.0), and the status code 200 (which lets the browser know that the client's request was successful and that data is forthcoming). The next line identifies what type of data will be transferred (here, we are simply sending text or HTML back to the client). The remaining part of the program iterates through a loop, using an associative array to capture the name of each environmental variable, its value, and output the results back to the client. The output of this program on an IIS 4.0 web server resulted in 38 environmental variables, with their names and values, being returned to the client as shown in Figure 3.4.

```
I = 0 SERVER_SOFTWARE = Microsoft-IIS/4.0
i = 1 INSTANCE_ID = 1
i = 2 PROCESSOR_IDENTIFIER = x86 Family 5 Model 7 Stepping 0, AuthenticAMD
i = 3 PROCESSOR_ARCHITECTURE = x86
i = 4 OS = Windows_NT
i = 5 GATEWAY_INTERFACE = CGI/1.1
i = 6 INCLUDE = C:\Program Files\Mts\Include
i = 7 REMOTE_ADDR = 127.0.0.1
i = 8 REQUEST_METHOD = GET
i = 9 WINDIR = C:\WINNT
i = 10 HTTP_ACCEPT = image/gif, image/x-xbitmap, image/jpeg, image/pjpeg,
application/vnd.ms-excel, application/msword, application/vnd.ms-powerpoint, */*
i = 11 HTTP_ACCEPT_LANGUAGE = en-us
i = 12 HTTPS = off
i = 13 HTTP_ACCEPT_ENCODING = gzip, deflate
i = 14 SERVER_NAME = 127.0.0.1
i = 15 PROCESSOR_LEVEL = 5
i = 16 OS2LIBPATH = C:\WINNT\system32\os2\dll;
i = 17 SERVER_PORT = 80
i = 18 PATH_TRANSLATED = C:\InetPub\wwwroot\cgi-bin\cgivars.pl
i = 19 SYSTEMDRIVE = C:
i = 20 SERVER_PORT_SECURE = 0
i = 21 LOCAL_ADDR = 127.0.0.1
i = 22 SERVER_PROTOCOL = HTTP/1.1
```

```
i = 23 SYSTEMROOT = C:\WINNT
i = 24 REMOTE_HOST = 127.0.0.1
i = 25 COMSPEC = C:\WINNT\system32\cmd.exe
i = 26 HTTP_USER_AGENT = Mozilla/4.0 (compatible; MSIE 4.01; Windows NT)
i = 27 PATH = C:\WINNT\system32;C:\WINNT;C:\Program Files\Mts;C:\MSSQL\BINN
i = 28 PROCESSOR_REVISION = 0700
i = 29 HTTP_CONNECTION = Keep-Alive
i = 30 NUMBER_OF_PROCESSORS = 1
i = 31 USERPROFILE = C:\WINNT\Profiles\jsheld
i = 32 COMPUTERNAME = COMPUTER2
i = 33 SCRIPT_NAME = /cgi-bin/cgivars.pl
i = 34 LIB = C:\Program Files\Mts\Lib
i = 35 PATH_INFO = /cgi-bin/cgivars.pl
i = 36 CONTENT_LENGTH = 0
i = 37 HTTP_HOST = 127.0.0.1
```

Figure 3.4. Output of the *cgivars.pl*. script.

As previously discussed, since every one of these values must be determined for each CGI program that is executed, there is a large amount of server overhead that must be performed. With this in mind, as well as the numerous security-related concerns associated with CGI programs, Javasoft introduced the concept of a servlet in April 1997. Servlets are essentially the server-side version of an applet: a small piece of Java™ code that is loaded by a web server and used to deal with client requests, much like CGI [Ref. 7].

The two central differences between servlets and CGI programs are:

- The servlet *init* method allows programmers to perform resource-intensive operations at startup that are common to all servlet invocations. For example, by having the *init* method make a database connection, the connection is made once and maintained through the servlet lifecycle. Each serlvet invocation (generated from a client request) then uses this existing connection to perform database operations, and thus, the amount of overhead involved in establishing a connection is reduced to the first occasion that the servlet is executed. CGI programs cannot do this, and 1,000 simultaneous form submissions that have to be parsed into a database would result in 1,000 separate database connections being generated.

- Servlet development extends the benefit of using the Java™ programming language. Because Java™ is platform-independent, so are servlets, and they can be run on any platform without recompilation or rewriting. CGI programs written in high-level languages such as C are not portable [Ref. 7].

Servlets are an effective substitute for CGI programs. They can process data that is submitted via an HTML form, allow collaboration between people by synchronizing requests to support systems such as

36

online conferencing, and pass requests to other servlets to load balance the amount of work that a system or servlet is performing.

C.   SUPPORTING SERVLET DEVELOPMENT

Java™ servlets are not part of the core Java™ language. Rather, they are an extension to it and are defined in two packages: *javax.servlet* and *javax.servlet.http*. Javasoft provides a servlet Software Development Kit (SDK) that contains all of the pieces necessary for implementing servlets on web servers. Java™ Development Kits (JDK) 1.1.6 and later contain the servlet packages but do not have the servlet server engine for running and testing your servlets. By downloading the Servlet SDK, you get this engine (*servletrunner.exe*, ~41K), API documentation, and sample support for Netscape, Microsoft, and the Apache web servers [Ref. 8]. More information on the Servlet SDK and download instructions are at

[http://www.javasoft.com/products/java-server/servlets/index.html#sdk.]

We used Windows NT Server 4.0 running IIS to develop our Java™ servlets (IIS is available free of charge from Microsoft and is bundled with their Windows NT 4.0 Option Pack (~27 MB)). IIS requires an Intel (Pentium 66 MHz or greater) or compatible system with 32 MB of memory (RAM) and 200 MB of available hard-disk space for proper installation. If you system meets or exceeds these specifications, you can download the option pack from:

[http://www.microsoft.com/windows/downloads/contents/Updates/NT4OptPk/default.asp].

After installing the software, we began developing servlets in two environments. First using NT's Disk Operating System (DOS) emulator, we made changes to the machine's environmental variables through the Control Panel. To do this, simply start the Control Panel, select System, then edit the environmental variables in the following manner:

- PATH - Add the absolute path to the bin directory of the JDK that you are using. The PATH statement enables Windows to find the executables Java™, javac, javadoc, etc… which you use when writing Java™ applications, from any current directory (You can always find out the current value of your PATH by typing "path" at the DOS prompt).

- CLASSPATH - Installation of the JDK should create and set this variable. You can determine if it is currently set by typing "set" at the DOS prompt. This lists all of the environment variables (CLASSPATH will not appear if it is not set). CLASSPATH

37

tells the Java™ virtual machine and other applications (which are located in the JDK bin directory) where to find the class libraries. CLASSPATH can indicate where a directory of class files are located, it can point to a zipped file or Java™ Archive (JAR) file containing class files, or it can refer to a specific class file. By default, the Java™ tools temporarily append the following to whatever CLASSPATH you have explicitly set in your startup file:

.;[bin]\..\classes;[bin]\..\lib\classes.zip

where [bin] is substituted by the absolute path to the JDK bin directory. Therefore, if you keep the bin and lib directories at the same directory level (that is, if they have a common parent directory), the executables will find the classes. You need to set the CLASSPATH only if you move *classes.zip* or want to load a different library (such as one you develop). For more information on how we configured these settings for our project development, see Table 4.1.

PATH	.;e:\jdk1.2beta3\bin;d:\VisualCafePDE\BIN;d:\VisualCafePDE\JAVA\BIN
CLASSPATH	.;e:\jdk1.2beta3\bin\classes.zip;e:\jsdk2.0\lib\jsdk.jar;d:\VisualCafePDE\BIN\COMPONENTS\SYMBEANS.JAR;d:\VisualCafePDE\JAVA\LIB\CLASSES.ZIP;d:\VisualCafePDE\JAVA\LIB;

Table 3.1. Project settings for PATH and CLASSPATH.

Making these changes allows the developer to use a text editor, such as *notepad.exe* or *edit.com*, to create, compile, debug, and run Java™ applications from the DOS prompt. This procedure is quite primitive when compared to the Rapid Application Development (RAD) tools that are available for Java™. Therefore, we decided to explore such a tool – namely, Symantec's Visual Café product (version 2.5). This Integrated Development Environment (IDE) is extremely powerful, provides a great deal of built-in support for applications and applets, and has many customizable settings that allow you to support other types of software development. In order to use Symantec's IDE for servlets, we had to modify the program's CLASSPATH (which is independent of the DOS CLASSPATH setting). The Café product automatically generates the CLASSPATH it needs to support its normal operation. By allowing the developer to append to that path, you can develop servlets by simply including the appropriate jar file – *jsdk.jar* (see Figure 3.5).

Jsdk.jar is part of the Servlet SDK distribution. You can view the contents of any jar file by typing "*jar -tf <jar file>*" at the DOS prompt. The archived jsdk file contains 27 class files:

SERVLET CLASS FILES	UTILITY CLASS FILES	OTHER CLASS FILES
HttpDate	Ascii	ServletConnection
HttpDateUpdater	Pool	ServletLoader
HttpInputStream	Queue	
HttpOutputStream HttpRequest	QueueFullException	
InvalidRequestException	QueueEmptyException	
HttpRequestLine		
MessageString		
MessageBytes		
MimeHeaders		
MimeHeadersEnumerator		
HttpSessionImpl		
SessionContextImplHttpResponse		
HttpResponse$OldCookieExpiry		
HttpRequestListener		
MimeHeaderField		
CacheControlHeader		
HttpServer		
HackURLConnection		
HttpServerHandler		
HttpServletConfig		

Table 3.2. *Jsdk.jar* class files.
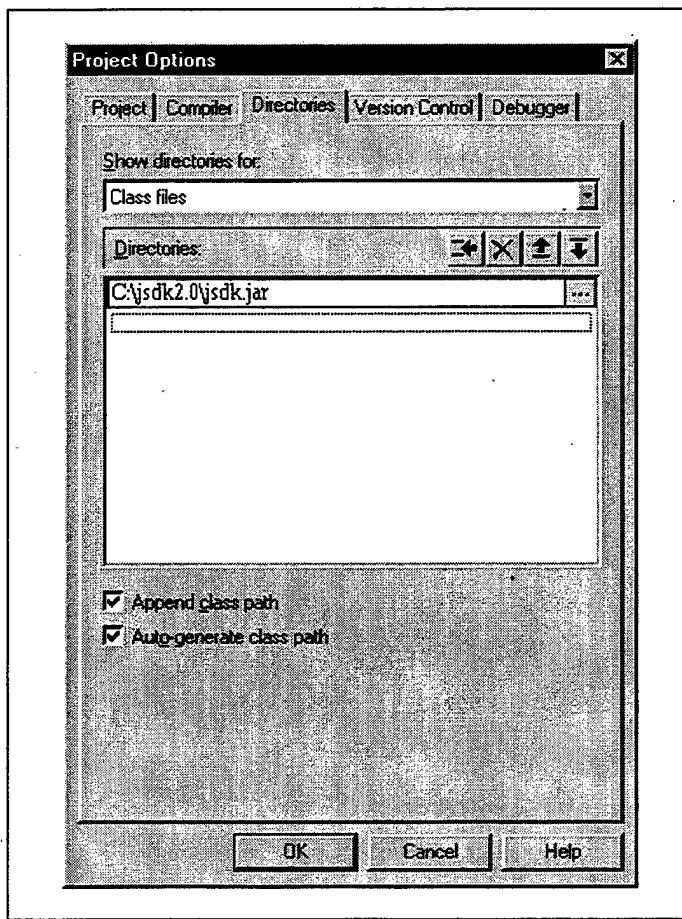
Figure 3.5. Modifying Café's CLASSPATH.

While Symantec's product did offer an extremely easy-to-use, self-explanatory interface and powerful tools for developing Java™

39

applications and applets, it did have some noticeable drawbacks. First and foremost, there is a fair amount of extraneous code generated when using Symantec's drag-and-drop Graphical User Interface (GUI) editor. This results in larger class files, and code which is not optimized for network performance. By manually editing the code, you can significantly reduce the size of the Java™ bytecode (class file) and reduce the amount of download time over network connections. Unfortunately, the process of editing code is cumbersome and takes a fair amount of time and reorganization of code that Symantec automatically generates. Other problems include:

- Two-way parsing between the GUI editor and underlying code does not work. When components are removed from the GUI designer, the underlying code is not completely changed, and when components are added in the code, they are not visible on the GUI designer.

- Sometimes, you may lose all GUI designer functionality. If this occurs, you can no longer use the drag-and-drop features of the GUI designer.

- When Café crashes (which occurs quite often), changes are lost since the last good compile and other computer components, such as the monitor and sound card, may not work properly. The end result may require rebooting the system to regain control and/or functionality of all computer hardware.

- Applet development works natively under Microsoft's Internet Explorer 4.0, however, using the applet in a Netscape browser requires a Java™ plug-in. There are also noticeable "appearance" differences (e.g. dialog boxes may have a background color in one browser but not the other).

D.   SURVEYSERVLET

Since servlets are faster and safer than CGI programs, we decided to use them to accept, process, and input the ACSA survey information into a SQL 6.5 database. The HTML front-end was designed using Microsoft Front Page 98, and serves primarily as the interface of choice for clients who are limited in what they can do by their computing resources. The HTML form also has the added benefit of allowing the client to avoid installation and configuration of the Java™ plug-in.

While most of the HTML is self-explanatory (you can find a complete copy of the HTML and the Javascript error-checking files in Appendices D and E), it is worth highlighting and explaining the meaning of at least one key HTML tag:

40

```
<form action="http://spitfire.avsafety.nps.navy.mil:8080/servlet/SurveyServlet" method="post"
name="survey" onsubmit="return check_form();">
```

The form tag is used to inform the browser that the HTML code
that follows is part of a form. What to do with that form is specified
by the action attribute. In this case, we inform the browser to take
the contents of the form and send them to port 8080 of the Uniform
Resource Locator (URL) *spitfire.avsafety.nps.navy.mil.* Unbeknownst to the
client, a servlet server is listening on that port, and will invoke the
SurveyServlet.

The method of sending the information is via the POST method. The
POST method transfers information via HTTP headers, and the information
is encoded and put into a header called QUERY_STRING for further
processing [Ref. 9]. It is worth noting that you can only post
information to files that are executable and that can read the
QUERY_STRING; otherwise, you will receive an error [Ref. 9]. We could
have sent the data via the GET method, however, there are some serious
limitations to doing this that should be noted:

- The GET method encodes data using a scheme called URL
  encoding. In this schema, name/value pairs are joined with
  equal signs and different pairs are separated by the ampersand
  [Ref. 9].

  e.g. name1=value1&name2=value2&name3=value3

  Spaces are removed and replaced with the "+" character, and
  any other nonalphanumeric characters are replaced with a
  hexadecimal value according to RFC 1738 [Ref. 9]. The GET
  method sends the encoded user information appended to the URL
  that processes the request. The page and the encoded
  information are separated by the "?" character, as follows:
  http://www.example.com?name1=value1&name2=value2&name3=value3

  With lots of information to send to the server, this can be a
  quite long, ugly string. More importantly, however, it is a
  security hazard because if the page is printed or someone
  other than the sender can view the page, then the
  confidentiality of the information is compromised.

- The GET method has a size limitation of 1,024 characters.

The remaining two attributes of the form are its name and the
Javascript function to execute when the form is submitted.

E.   SERVLET ARCHITECTURE

The *javax.servlet* package provides all of the interfaces and classes needed for writing servlets. The central abstraction in the Servlet Application Programming Interface (API) is the Servlet interface, which is implemented either directly or by extending a class that already implements it [Ref. 10].
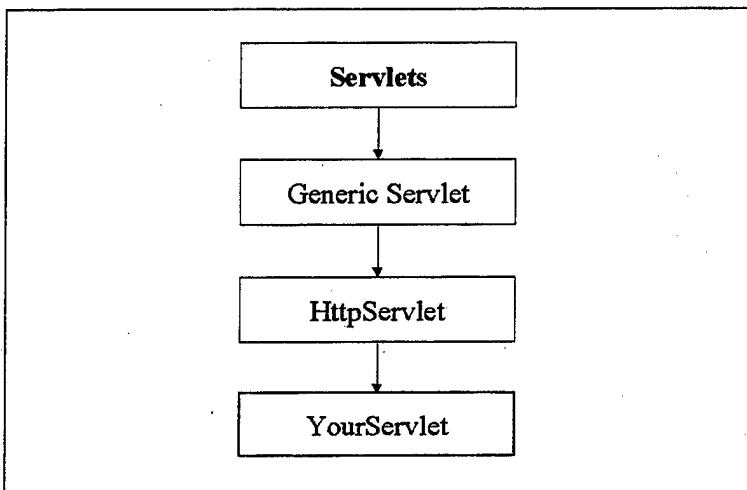
Figure 3.6. Servlet Architecture.

The Servlet interface declares many methods but does not provide implementations for them – this task is left to the programmer, who may choose to override the behavior of certain methods (e.g. how a GET or POST operation is handled). The inheritance of methods by your servlet is graphically represented by Figure 3.6.

The Servlet API encapsulates the communication to and from the client via a request and response object. Specifically, the *ServletRequest* handles all communication from the client to the server, and the *ServletResponse* provides a means communicating from the servlet back to the client [Ref. 10]. A look at the *SurveyServlet* will provide a solid example of how servlets work, what events occur, and when these events are executed.

```
1: import java.io.*;
2: import java.sql.*;
3: import javax.servlet.*;
4: import javax.servlet.http.*;
5: import file_utilities.*;

6: public class SurveyServlet extends HttpServlet
7: {
```

Figure 3.7. *SurveyServlet.*

42

Lines 1-5 of Figure 3.7 are used to import the various libraries that the servlet code will use. Class *SurveyServlet* extends, or inherits, the methods and data members of HttpServlet. Hence, *SurveyServlet* gets a lot for free, but in order to know what it is that *SurveyServlet* already has by virtue of the inheritance, we have to look at the *HttpServlet* class. If you have obtained the Servlet SDK, you will find the source code for *HttpServlet* (contained in the file *HttpServlet.java*). There is a brief explanation of what the class is used for:

> An abstract class that simplifies writing HTTP servlets. It extends the <code>GenericServlet</code> base class and provides an framework for handling the HTTP protocol. Because it is an abstract class, servlet writers must subclass it and override at least one method [Ref. 11].

and the documentation goes on to inform the reader what *HttpServlet* methods are generally overridden and what these methods do. Looking through the *HttpServlet* source code, we find pre-defined implementations for the following methods:

- *doGet*:

```
protected void doGet (HttpServletRequest req, HttpServletResponse resp)
    throws ServletException, IOException {
    resp.sendError (HttpServletResponse.SC_BAD_REQUEST,
                "GET is not supported by this URL");
}
```

- *doHead*:

```
private void doHead (HttpServletRequest req, HttpServletResponse resp)
    throws ServletException, IOException {
    NoBodyResponse response = new NoBodyResponse(resp);
    doGet (req, response);
    response.setContentLength ();
}
```

- *doPost*:

```
protected void doPost (HttpServletRequest req, HttpServletResponse resp)
    throws ServletException, IOException {
    resp.sendError (HttpServletResponse.SC_BAD_REQUEST,
                "POST is not supported by this URL");
}
```

- *doPut*:

```
protected void doPut (HttpServletRequest req, HttpServletResponse resp)
    throws ServletException, IOException {
    resp.sendError (HttpServletResponse.SC_BAD_REQUEST,
                "PUT is not supported by this URL");
}
```

- *doDelete*:

```
protected void doDelete (HttpServletRequest req, HttpServletResponse resp)
            throws ServletException, IOException {
    resp.sendError (HttpServletResponse.SC_BAD_REQUEST,
            "DELETE is not supported by this URL");
}
```

As mentioned earlier, the GET and POST actions are the most common operations that are generated from form submissions on HTML pages. Since *SurveyServlet* inherits methods from *HttpServlet* and is invoked via a POST operation from the HTML page, it would be wise to override the default behavior of the method *doPost*. If we fail to do this, an error will be returned to the client indicating that "POST is not supporting by this URL."

Before looking at this method, however, recall that earlier, when we talked about advantages of servlets over cgi programs, we said that servlets can perform resource-intensive operations once and then share these resources among all servlet invocations. This implementation is extremely beneficial because it greatly reduces the amount of overhead involved in performing database connections. Whereas CGI programs must continually create and destroy such connections, servlets perform the connection only once in their *init* method. Multiple servlet invocations can share the connection resources amongst themselves, and requests or queries are queued as they are made. When a servlet is destroyed, it can release the resources it is holding by performing the necessary actions in the method *destroy*.

Because *SurveyServlet* has to communicate with two databases, it has an *init* method, and it is here that our database connections are made:

```
1:  //-----------------------------------------------------------------------
2:  //Function:     init()
3:  //Parameters:   config - a servlet configuration object; this object is not
4:  //                   manipulated in the code; rather, the superclass is called,
5:  //                   and the object is manipulated by that class
6:  //Return type:  None
7:  //Purpose:      Initializes the servlet. Resource-intensive procedures, such as
8:  //                   making a database connection, as well as shared resources, such
9:  //                   as the WriteToFile object, which stores a backup of the survey
10: //                   information, is done here.
11: //-----------------------------------------------------------------------
12:    public void init(ServletConfig config) throws ServletException {
13:        super.init(config);

14:        //shared resources of all servlets - a connection to the database, and
15:        //a thread that saves a backup of form data to a plaintext file in the
16:        //logs directory
```

```
17:     try {
18:       Class.forName("connect.microsoft.MicrosoftDriver");
19:       con1 = DriverManager.getConnection(url1, "sa", "");
20:       con2 = DriverManager.getConnection(url2, "sa", "");
21:       stmt1 = con1.createStatement();
22:       stmt2 = con2.prepareStatement("SELECT * FROM Num_List WHERE Unit_ID = ?");
23:       fileop = new file_utilities.WriteToFile();
24:     }
25:     catch (java.lang.ClassNotFoundException e) {
26:       System.out.println(e.toString());
27:       generalError = true;
28:     }
29:     catch (java.sql.SQLException e){
30:       generalError = true;
31:     }
32:   }//end init()
```

Figure 3.8. Servlet *init* method – performing resource-intensive operations.

The *init* method takes a *ServletConfig* object and sends it to the superclass' (*HttpServlet*) *init* method for initialization. A *ServletConfig* object is used for servlet configuration, and once the superclass has initialized it as appropriate, there is nothing more that *SurveyServlet* needs to do with it. Line 17 is where *SurveyServlet* loads a JDBC driver and uses JDBC to make connections to databases at url1 and url2 (see Appendix F). Suffice it to say that connections are made to both databases (if not, then exception handling will take care of the errors). Once the connections are made and JDBC *statement* objects are initialized (lines 21 and 22), the *init* method is complete. The servletrunner (discussed in section F of this chapter), once informed of where the *SurveyServlet* class file is, runs and waits for a client to invoke the servlet. The very first time information is sent from the HTML form to the servlet server, the servletrunner initializes the *SurveyServlet* by executing its *init* method then executes the remainder of the code. On this first invocation, connections are made to both databases, and there appears to be a great amount of overhead associated with this operation. By observing subsequent invocations, you will see that this delay occurs only once – during the very first servlet invocation – and the response time is extremely fast thereafter.

To process the information from the HTML form, *SurveyServlet* explicitly overrides the *doPost* method. After the *init* method is complete, execution continues with the *doPost* method as outlined in Figure 3.9:

45

```
1:   //------------------------------------------------------------------
2:   //Function:    doPost()
3:   //Parameters: request - an HttpServletRequest object that allows us to
4:   //                grab the values the client sent the servlet
5:   //                response - an HttpServletResponse object that allows us to
6:   //                return a response back to the client
7:   //Return type: None
8:   //Purpose:    Makes several calls to utility functions that operate on the
9:   //                client's input - the client is redirected to a different page
10:  //                depending on the success or failure of the POST operation.
11:  //------------------------------------------------------------------
12:    public void doPost(HttpServletRequest request, HttpServletResponse response)
13:       throws ServletException, IOException {

14:       entryError =  false;
15:       boolean goodID = false;

16:       if (!generalError){
17:       .  parseFormData(request);

18:       goodID = checkSurveyNumbersDatabase();
19:       if (goodID){
20:         makeDatabaseEntry(response);

21:        if (!entryError){
22:          fileop.start();
23:          fileop.write(clientdata, ipAddress, host);
24:          try {
25:             response.setContentType("text/html");
26:             response.sendRedirect("http://spitfire.avsafety.nps.navy.mil/ThankYou.html");
27:          }
28:          catch (java.io.IOException e){
29:             System.out.println(e.toString());
30:          }
31:        }
32:       }
33:      else {
34:        response.sendRedirect("http://spitfire.avsafety.nps.navy.mil/NumberError.html");
35:      }
36:    }
37:    else
38:      response.sendRedirect("http://spitfire.avsafety.nps.navy.mil/GeneralError.html");

39:   }//end doPost()
```

Figure 3.9. Overriding the *doPost* method to handle HTML form input.

The *doPost* method receives two objects:

- an *HttpServletRequest* object that allows us to grab the values of the information the client sent us, and

- an *HttpServletResponse* object that enables the servlet to communicate with the client either by sending back raw HTML code, or redirecting the client to another page.

46

These objects' methods, their methods' signatures, and what these
methods do are documented at Javasoft's web site [http://javasoft.com]
Line 16 checks to make sure there are no general errors (such as
failure to connect to a database), and if there are none, we proceed to
parse the form's data. We have a method in *SurveyServlet* that does this
for us – *parseFormData* (see Figure 3.10) – and it requires the
HttpServletRequest object that was initially received by our *doPost*
method.

```
40. //------------------------------------------------------------------------
41. //Function:    parseFormData()
42. //Parameters:  request - an HttpServletRequest object that allows us to
43. //                grab the values the client sent the servlet
44. //Return type: None
45. //Purpose:     Retrieves the form information that was sent to the servlet.
46. //------------------------------------------------------------------------
47.   private void parseFormData(HttpServletRequest request) {

48:       ipAddress = request.getRemoteAddr();
49:       host = request.getRemoteHost();

50:       //form parameters - form is made up of bio_data and questions
51:       squadron_type = new String(request.getParameter("community"));
52:       unit_number = new String(request.getParameter("squadron_number"));
53:       survey_number = new String(request.getParameter("survey_number"));

54:       java.lang.String bio_data[] = {"community", "rank", "designation", "flight_hours", "type_hours",
55:                                     "department_head", "status", "service", "location"};

56:       java.lang.String questions[] = new String[NUM_SURVEY_QUESTIONS];

57:       for (int ix=0; ix<NUM_SURVEY_QUESTIONS; ix++)
58:         questions[ix] = "question" + Integer.toString(ix+1);

59:       for (int jx=0; jx < clientdata.length; jx++){
60:         if (jx < NUM_BIO_QUESTIONS)
61:           clientdata[jx] = request.getParameter(bio_data[jx]);
62:         else
63:           clientdata[jx] = request.getParameter(questions[jx-NUM_BIO_QUESTIONS]);
64:       }

65:   } //end parseFormData()
```

Figure 3.10. *parseFormData* – getting the client's input.

The request object is used on lines 48 and 49 to get the Internet
Protocol (IP) address of the client that invoked the servlet, as well
as the host name of the client's machine. These values are saved for
later use in a log file. Lines 51-53 store the values from the form's
community, squadron number, and survey number fields. We obtain these
values by using the request object's *getParameter* method. This method

takes a string (the field from the form you want to evaluate) and
returns a string containing the lone value of the specified value or
null if the field does not exist.

The clientdata array is used to hold values for the biographical
data, as well as the client's responses to 61 questions. Once
processing is complete, we return to line 18 of the *doPost* method.
Because the ACSA survey is only intended for military personnel use, we
need some manner of identifying whether the client that submitted
information to us is a valid participant in the survey.

In order to implement some kind of security check, we opted to
create a database that stored a range of valid survey numbers for each
squadron. The function *checkSurveyNumbersDatabase* performs this task. If
the survey number is not valid for the squadron, the client is
redirected to the *NumberError.html* page. If the survey number already
exists for that squadron, a *SQLException* occurs and the method
handleSQLException takes over:

```
66. //------------------------------------------------------------------------
67. //Function:    handleSQLException()
68. //Parameters: response - an HttpServletResponse object that allows us to
69. //              return a response back to the client
70. //              ex – an SQLException
71. //Return type: None
72. //Purpose:     Generates HTML code that indicates to the user what SQL Error
73. //              was generated.
74. //------------------------------------------------------------------------
75.   private void handleSQLException(HttpServletResponse response, java.sql.SQLException ex){

76.     java.lang.String errorMessage = new String();
77.     while (ex != null){
78.       errorMessage += ("<br>Message: " + ex.getMessage() + "<br>SQL State: " + ex.getSQLState()
79.             + "<br>Error Code: " + ex.getErrorCode() + "<br>");
80.       System.out.println("Error executing SQL statements. Transaction not entered.\n");
81.       System.out.println("--------------------------------------------------------");
82.       ex = ex.getNextException();
83.     }

84.     try {
85.       PrintWriter out;
86.       response.setContentType("text/html");
87.       out = response.getWriter();
88.       out.println("<html><head><title>Error!</title></head><body bgcolor=#000000 link=#00FFFF>"
89.             + "<p><font color=#FFFF00 size=4>A problem was encountered while " +
90.             "trying to process your input. Possible causes for this are listed " +
91.             "below. If you have additional questions as to why your input was " +
92.             "not accepted, you may contact the webmaster at </font>" +
93.             "<big><a HREF=mailto:spitfire.avsafety.nps.navy.mil>AVSafetyCenter</a></big>"
94.             + "<font color=#FFFF00 size=4>, or you may complete the survey again " +
95.             "and try resubmitting it.</font></p><p><font color=#FF0000 size=4> " +
```

```
96.              errorMessage + "</big></big></font></p></body></html>");
97.    }
98.    catch (java.io.IOException e){
99.       System.out.println(e.toString());
100.   }
101. }//end handleSQLException()
```

Figure 3.11. Providing the method *handleSQLException* for client error notification.

   handleSQLException determines what the exact error was, saves the error message, error state, and error code in the string errorMessage, then prepares to respond to the client. The *PrintWriter* object (line 85) is what we'll use to write HTML back to the client. Since the response object already contains a writer we can use for this purpose (we could not create a new one for this purpose on our own), we simply call the method *getWriter*. This method returns a reference to the *PrintWriter* we need, and lines 88-96 use the *PrintWriter* object we have called "out" to send HTML back to the client.

   Assuming that the client is authorized to submit survey information, execution continues on line 20 of Figure 3.9 with a call to *makeDatabaseEntry*. This function contains the JDBC code necessary to enter the information into the SQL 6.5 database. If there are no SQL exceptions and the database is updated, the client is then redirected to the *ThankYou.html* page.

   Once the user is redirected, execution of *SurveyServlet* is complete. A graphical overview of *SurveyServlet* flow control is presented in Figure 3.12 for your reference:
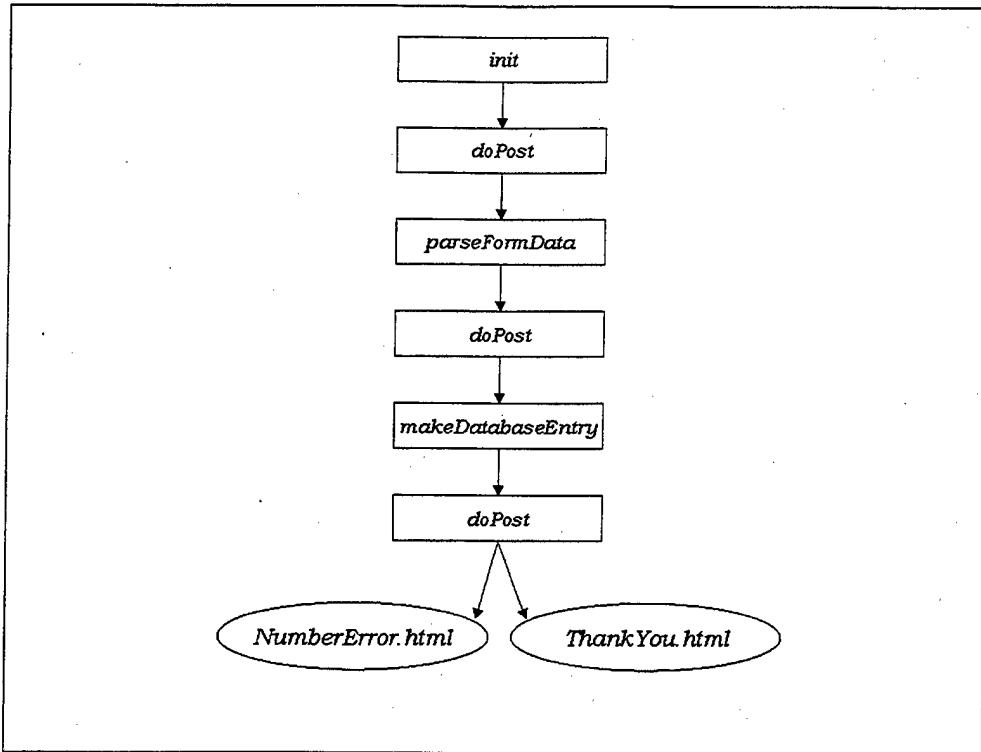
Figure 3.12. *surveyServlet* flow control.

F.   CONFIGURING THE SERVLETRUNNER UTILITY

Servletrunner is a multi-threaded, executable program bundled with the Servlet SDK that handles requests for servlets. Because it is multi-threaded, servletrunner can handle requests for multiple servlets simultaneously [Ref. 12].

After you have developed and compiled your servlet code, you must tell the servletrunner, prior to starting it, where the code is located. To do this, you modify the file *servlet.properties*. This ASCII text file uses key-value pairs to specify the name of the servlet code and if the servlet has any initial arguments that need to be passed to it. Configuring the servletrunner for *SurveyServlet*, we added:

servlet.*SurveyServlet*.code = *SurveyServlet*

This line informs the server of two things: there is a servlet, named *SurveyServlet*, that may be requested, and if it is requested, the code for the servlet can be found in a file named *SurveyServlet.class*. In general, where servlet code resides is specified using the following syntax:

servlet.*name*.code = *classfile,*

where *name* is the name of the servlet as referenced from the HTML page, and the *classfile* is the corresponding Java™ class bytecode to execute when the servlet is requested. In this manner, it is not necessary for the name of the classfile and servlet to be identical, however, it does add clarity to servlet configuration when they are the same.

To send initial arguments to the servlet, use the following syntax:

```
servlet.name.initargs=\
            <arg 1>=<value 1>,\
            <arg 2>=<value 2>,\
                    .
                    .
                    .
            <arg n>=<value n>
```

The initial argument property must be a single logical line and requires the backquote syntax to allow the property to span multiple lines in the file [Ref. 12]. Once you are done modifying the "*servlet.properties*" file, you can start the server by typing "*servletrunner*" at the command prompt.

If successful, you should see something like this:

Figure 3.13. Starting the servletrunner.

By invoking the servletrunner at the DOS prompt with "*servletrunner /?*" or "*servletrunner –help*", you can obtain a list of configurable settings:

```
Usage: servletrunner [options]
Options:
-p port the port number to listen on
-b backlog the listen backlog
-m max maximum number of connection handlers
```

```
-t timeout connection timeout in milliseconds
-d dir servlet directory
-r root document root directory
-s filename servlet property file name
-v verbose output
```

The default port settings are shown in Figure 3.13.

With this information, you should now be able to write, run, and test your own Java™ servlets in no time at all. If you make changes to your servlet code and want those changes to take effect, you will need to stop the servletrunner and restart it.

G.   CONCLUDING REMARKS

Servlets are extremely easy to write and use and are supported by an extensive and well-documented API. Using servlets gives you other benefits as well: the stability of the Java™ programming language requires that all possible exceptions, or programming errors, that could occur during the course of execution are appropriately handled. This means that your program, if properly written, should not crash. Additionally, your servlets inherit all of the limitations posed by the Java™ security model, so unlike cgi programs, malicious hackers cannot exploit your programs for their benefit. Finally, RAD tools significantly reduce development time, allowing you to deploy servlet solutions in a matter of hours. All of these advantages strongly suggest that servlets are an acceptable and better substitute for CGI.

IV. MICROSOFT'S ACTIVE SERVER PAGES (ASP)

A.   ASP BACKGROUND

Microsoft's Active Server Pages (ASP) technology was originally introduced with the release of the company's third version of Internet Information Server (IIS) and its Personal Web Server (PWS) software. These freeware web servers for the Windows NT/98 and 95 platforms can be downloaded from [http://www.microsoft.com]. If you are using another vendor's web server, you can get the same ASP functionality by installing Chili! ASP. This product is the equivalent of Microsoft's ASP but extends ASP operability to Netscape, Lotus, IBM and other web servers [Ref. 13].

ASP is a fairly new technology that is only a small part of Microsoft's larger *Active Platform* Internet strategy. According to Microsoft, the *Active Platform*

> ...is an open, standards-based software architecture
> for delivering rich content and line-of-business
> applications over the Internet and intranets. It
> combines the power of PC and network computing to
> link people with information, anywhere, any time
> [Ref. 14].

Unfortunately, this tells us little about what the *Active Platform* idea really is. As you read further into Reference 2, it becomes clearer that the *Active Platform* is a very broad concept that consists of technologies related to HTML, scripting, and ActiveX or JavaBean™ components. If the HTML, scripts, and components are executed on the client's machine, then they are part of what is called an *Active Client;* similarly, should the opposite occur, then we have what is called an *Active Server*. The *Active Client, Active Server,* and reusable components are all part of the *Active Platform* strategy.

Aside from being a new buzzword to add to many others, the *Active Platform* strives to promote language transparency and component reuse by adhering to approved standards. Vendors that create products based on these standards give developers invaluable tools that allow them to create rich, content-aware, platform-independent web sites almost effortlessly. So where does ASP fit into all of this?

ASP is an open compile-free application environment that allows developers to use HTML, scripting, and components to create dynamic and

powerful Web-based sites. This makes ASP an indispensable part of the *Active Platform* strategy.

To understand how ASP works, you should know the following: ASP is nothing more than a component that is used by a web server for processing special files. When a client requests a file that has an *.asp* extension, all scripting code in the file is processed on the server, and the server returns the processed output in the form of HTML back to the client. ASP allows developers to create programmable web pages that mix HTML and ODBC database operations without any knowledge of CGI, JavaScript, PERL, ActiveX, or ISAPI [Ref. 15].

So why use ASP? Apart from the reasons cited above, there are a number of additional advantages to using ASP technology:

1. ASP is extremely easy to learn. Key points to remember when developing with ASP are:

    - ASP uses the special scripting delimiters *<% %>* and *<%= %>*. The former pair of delimiters is used to enclose ASP script blocks, while the latter is used solely for the purpose of evaluating ASP expressions.

    - ASP is comprised of five built-in objects (*Application, Session, Request, Response,* and *Server*) with many predefined methods and properties.

    - ASP provides native support for both VBScript and Jscript. Effectively and efficiently using either scripting language allows you to develop extremely robust web applications.

2. ASP development does not require any compilation. Prior to ASP, interactive web sites that used CGI programs or ActiveX components required an Integrated Development Environment (IDE) such as Microsoft's Visual C++ in which they could first be compiled, and then the executable code was placed in the appropriate subdirectory on the web server.

3. ASP allows you to protect your code. Previously, client-side scripting was standard, but how the script behaved could be determined by viewing the HTML source of the web page. By executing the script on the server, the client can no longer see the code or interpret its meaning and proprietary algorithms can be protected without fear of compromise.

B. **THE ASP OBJECT MODEL**

The five built-in objects that ASP provides are special because they do not require any configuration or initialization before you can use them in your scripts. If you are familiar with the concepts of

Object-Oriented Programming (OOP), you should have no problem understanding how the five ASP objects work and how to access their methods and properties.

This section gives you a brief overview of the ASP object model and how each object is used in a typical ASP application. Once you have read through this introductory material, we move on to discuss the ActiveX Data Objects (ADO) and how they are used in conjunction with ASP to create database-driven web pages.

1. *APPLICATION* Object

You can use the *Application* object to share information among all users of a given application [Ref. 16]. Variables stored in the *Application* object have application scope, persist throughout the duration of application execution, and are accessible to subsequent web pages. These variables are defined in the file *global.asa*, which can also include code for the events *Application_OnStart*, *Session_OnStart*, *Application_OnEnd* and *Session_OnEnd*.

The "firing sequence" of events begins when the web server processes the very first request for an ASP-based application. The order in which events are processed is as follows:

1. *Application* objects are created when the first client connects to an ASP-based application and requests a session.

2. When the *Application* object is created, the server looks for the *global.asa* file, and if the file exists, executes the script in the *Application_OnStart* event handler. If the *global.asa* file has been modified since the last time it was loaded, the web server loads the new, updated copy.

3. Once the code for the *Application_OnStart* event handler is executed, a *Session* object is created and program control moves to the *Session_OnStart* event handler.

The sequence for destroying these objects occurs in the exact reverse order, and the *Session* object is disposed of prior to the destruction of the *Application* object:

4. When the *Session* object times out or the *Session.Abandon* method is called, the *Session_OnEnd* event in the *global.asa* file is triggered. Once this code is done executing, the *Session* object is destroyed.

5. When the server is shut down, the *Application_OnEnd* event is triggered, code in the event handler is executed, and the *Application* object is destroyed.

55

The following table highlights the syntax, methods, and events of the *Application* object:

SYNTAX:	*Application.method*
METHODS: *Lock*	Prevents multiple clients from concurrently accessing *Application* objects.
Unlock	Allows other clients access to *Application* objects.
EVENTS: *Application_OnStart*	Contains code that is executed before the very first new session is created. Of the five ASP objects,, only the *Application* and *Server* objects can be used in script code. For this event handler.
Application_OnEnd	Contains code that is executed when the application quits. This event handler is executed just after the *Session_OnEnd* event is processed.

Table 4.1. The *Application* object.

2.   *SESSION* Object

Immediately after the *Application* object is created, ASP starts a "session" for the client. This session is different for every user visiting your site, and it is by using the *Session* object that ASP can differentiate one user from all others. Session information is explicitly set in the *Session_OnStart* event handler of the *global.asa* file, and its contents are available as long as the session is kept active. As a user jumps from one web page to another, the information in the *Session* object persists. It is only after the session expires, or it is explicitly abandoned, that the web server calls the *Session_OnEnd* event handler and the object is destroyed.

The *Session* object can be used for any number of reasons, but it is especially well suited for storing user preferences or establishing database connection parameters. The *Session* object is often used for the latter purpose, as it allows you to have all the information necessary for opening resource-intensive data source connections when needed rather than during the entire duration of an ASP application. By using multiple, independent *global.asa* files, you have the flexibility to dynamically change both *Application* and *Session* object information.

Unfortunately, session information can only be maintained for browsers supporting cookies. A cookie is used to coordinate the user's session id [Ref. 17]. If clients do not allow cookies, or they are using older browsers that do not support cookies, you cannot use the

56

Session object. Another subtle point you should note about the *Session* object is that session data is stored on the server, not the client's browser. You should take care to optimize your use of session parameters, otherwise, you may place an extraordinarily heavy load on your server.

Table 4.2 gives an overview of the syntax, properties, and methods of the *Session* object:

SYNTAX:	*Session.property	method*
PROPERTIES: *SessionID*	Returns a unique session identification number for the user. The number is unique only as long as the web server is running. Once it is stopped, then restarted, session id numbers may be repeated.
Timeout	The time, in minutes, for the session state of the ASP application.
METHODS: *Abandon*	Explicit call to destroy a *Session* object and release its resources.

Table 4.2. The *Session* object.

3. *REQUEST* Object

The *Request* object is used to retrieve values that the client passes to the web server during an HTTP request [Ref. 18]. The values that are passed not only include form input, but also environmental variables, certificate information, and cookies. Information is stored in various collections within the *Request* object, and these collections can be used to access the client information. ASP provides five collections, as shown in Table 4.3.

COLLECTION	PURPOSE
ClientCertificate	Holds the value of fields stored in the client certificate.
Cookies	Holds the value of cookies.
Form	Holds form values processed via the *POST* method.
QueryString	Holds form values processed via the *GET* method.
ServerVariables	Holds environmental variables, such as *"REMOTE_HOST"*, *"REQUEST_METHOD"*, *"SERVER_NAME"*, etc… (For a complete listing of environmental variables supported by IIS, see Chapter III, Figure 3.4)

Table 4.3. The five *Request* object collections.

When using these collections, there are several important points to remember:

- If you request a variable that is not in one of the preceding five collections of Table 4.3, the *Request* object returns *EMPTY.*

- When you use ASP to process form input, the variables that are referenced in the *Form* or *QueryString* collections can be found embedded in the code for the HTML form object. With complex, multi-page forms that have to be processed, it is a wise idea to have a copy of the HTML code next to you when developing ASP scripts to process your data.

- The proper syntax to access a collection is *Request.* [*Collection*] (*variable*)

4.   *RESPONSE* Object

Now that you have client information and can process it in whatever manner you so desire, the next logical step is to provide you with some means of communicating back to the client. ASP gives you the *Response* object for this purpose. Of all ASP objects, the *Response* object is used the most and it supports the properties and methods outlined in Table 4.4.

SYNTAX:	*Response.collection	property |method*
COLLECTION: *Cookies*	Allows you to set cookie values if the client's browser accepts them.
PROPERTIES: *Buffer*	Boolean value indicating whether the page output is buffered.
ContentType	Specifies the *HTTP MIME* content type that is being returned to the client.
Expires	Specifies the length of time, in minutes, that a page can be cached on a browser; if set to zero, the page will not be cached.
ExpiresAbsolute	Lets you specify the date and time that a page will expire. If no time is specified, the page expires on midnight of that day. The date and time must be in Greenwich Mean Time (GMT) format as outlined by Request For Comments (RFC) 1123.

	Status	Allows you to indicate what the server status is. Common values and their meanings are: • 200: The request was successful and the data is forthcoming. • 301: The requested content has permanently moved to a new location. • 302: The requested content has temporarily moved to a new location. • 404: The requested content does not exist on the server. • 500: The server encountered an internal error while processing your request.
METHODS:	AddHeader	Allows you to create your own HTTP header and send it to the client.
	AppendToLog	Adds a string to the end of the web server log for this request.
	BinaryWrite	Write information to the HTTP output stream without any special character conversion.
	Clear	Erases any buffered HTML output.
	End	Stops processing the ASP application and returns the results up to the call to *End*.
	Flush	Sends buffered HTML output immediately to the HTTP output stream.
	Redirect	Sends a redirect message to the client's browser, causing the browser to open a new URL.
	Write	Writes a variable to the HTTP output stream.

Table 4.4. The *Response* object.

5.   *SERVER* Object

The *Server* object is probably the least used ASP object. This object has only one property – *ScriptTimeOut* - that allows you to set the value for when script processing will stop. Any script that runs longer than the time specified by *ScriptTimeOut* is assumed to have misbehaved and will be terminated to free system resources [Ref. 9]. The default value for this property is 90 seconds.

Of the object's four methods (*CreateObject, HTMLEncode, MapPath,* and *URLEncode*), *CreateObject* is used the most. This method creates server (ActiveX) components that are part of a web-based ASP application. These components, which are based on the Microsoft Component Object Model (COM) and Distributed Component Object Model (DCOM), provide key functionality for common web page tasks (e.g. database access, counting page accesses, etc…). They exist so that you do not have to create or

recreate the code to perform these tasks. Of the many components included with ASP, the most frequently used ones are listed in Table 4.5.

COMPONENT	PURPOSE
Database Access (ADO)	Provides access to databases from your web application.
Ad Rotator	Alternates a series of images.
Content Rotator	Automates the rotation of HTML content strings on a web page.
Browser Capabilities	Allows you to determine what the client's browser's capabilities are.
File Access	Allows you to retrieve and manipulate files on the server.
Content Linking	Allows you to provide logical navigation through the *.asp* files that are part of an ASP application.
MyInfo	Keeps personal information about the web site administrator.
Tools	Provides methods for random number generation, checking for the existence of a form, etc…
Page Counter	Counts and displays the number of times a web page have been requested.

Table 4.5. ActiveX components included with ASP [Ref. 19].

With the *Server* object, you can easily use one of these components, develop your own, or use someone else's. Whatever the case, the syntax for doing so is always

Set variablename = Server.CreateObject(registeredobject)

C.   ACTIVEX DATA OBJECTS (ADO) OBJECT MODEL

Over the years, Microsoft has created a number of object models that you can use to access data sources. The ADO is the most recent model, and provides an interface to the OLE DB SDK [Ref. 20]. A factor in considering which model to use is how reliable and durable the model is, as well as what kinds of support the manufacturer provides for it. Microsoft has made it publicly known that the ADO will replace the company's older data-access models (such as Data Access Objects (DAO) and Remote Data Objects (RDO)). Figure 4.1 illustrates some of the differences between these models. For example, while DAO and RDO offer access support to SQL data through ODBC calls, you will notice that these models cannot interact with non-SQL or legacy data sources.

Another advantage of transitioning now to ADO programming is that ADO capabilities will be expanded. Not only will ADO allow developers to access relational data providers, but there is also continuing work in supporting access to non-relational ones as well.

The ADO are used with many of Microsoft's products. While it is certainly an integral part of ASP, you can also find ADO support in other products such as Visual Basic and Visual C++. Once you have your machine configured for ADO support (you can do this by installing either of the previously mentioned software products, Visual InterDev, IIS, or the OLE DB SDK), you can create database-driven web pages.

The ADO object model is relatively simple and consists of four primary objects as illustrated in Figure 4.2. We will briefly describe how these objects are used, some of their methods and properties, and how you might use them in your ASP applications. For a complete, well-documented reference to the ADO Application Programming Interface (API), you should go to [http://www.microsoft.com/data/reference/ado2/default2.htm] or [http://www.microsoft.com/data/reference/ado2.htm].

The major ADO objects, collections, and their purpose are shown in Table 4.6.

Object	Purpose
Connection	Creates a connection to a data provider.
RecordSet	Creates a set of records from a query. You can move forward and backwards through recordsets (cursors).
Command	Points to SQL strings, stored procedures, or action queries that you can execute.
Errors	Collections of errors that occur from a failed data-access attempt.

Table 4.6. ADO objects [Ref. 21].
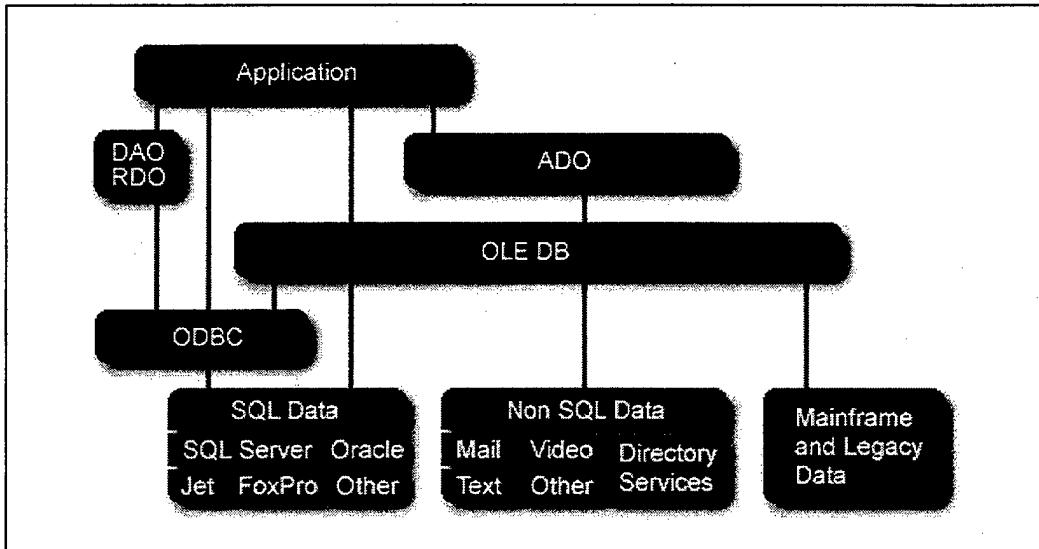
Figure 4.1. Comparison of ADO, DAO, and RDO [Ref. 22].

61
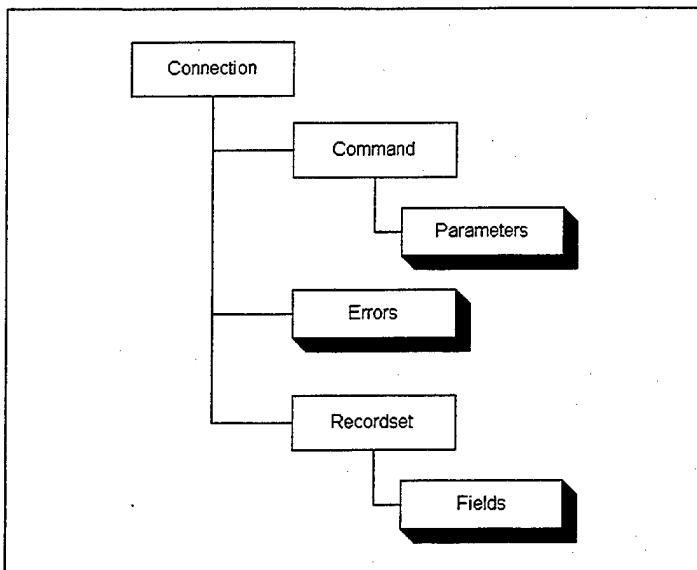
Figure 4.2. The ADO object model [Ref. 23].

1.   *CONNECTION* Object

All ADO objects can be created independently, but they cannot
communicate with a data source unless they go through the *Connection*
object. The *Connection* object allows you to connect to a data source,
perform insert, delete, and update operations, and execute SQL queries.
After you have successfully created a data source on your machine (how
to do this is discussed in Chapter V), you can use the source's name to
let the *Connection* object know what database on the server you intend to
interact with. To create a *Connection* object, use the following syntax:

Set variablename = Server.CreateObject("ADODB.Connection")

Some of the more frequently used collections, methods, and properties
of a *Connection* object include:

Open	Opens a network connection to a data source.
Close	Closes the network connection.
ConnectionTimeOut	Sets the maximum value that ADO will wait when attempting to make a connection to a database. The default value is 15 seconds.
Errors	Collection that allows you to examine errors that occur when you attempt to interact with a data source.
Execute	Executes a SQL statement on the data source.

Table 4.7. Common collections, methods, and properties of the *Connection*
object.

2. *RECORDSET* Object

When you perform a query on a database using SQL, you need some means of retrieving the results produced from that the query. A *RecordSet* (or cursor) object allows you to do this. With a *RecordSet*, you can navigate through your results one row, or tuple, at a time. As you examine each row, you may look at all columns or fields of data, or you may choose to be selective in what you examine. What you should realize about the ADO *RecordSet* object is that it is not a composite object, i.e. it is composed of Field objects that represent each column of data in your row. You can create a *RecordSet* in the following manner:

Set *variablename* = Server.CreateObject("*ADODB.RecordSet*")

There are properties of the *RecordSet* that you can set at the time you create it. These properties allow you to specify what type of functionality you want your *RecordSet* to have. The four different types of *RecordSets* are:

1. *Dynamic*: Allows you to view additions, changes, and deletions by other users, and you can freely jump to different locations (rows) in the *RecordSet*.

2. *Keyset*: Behaves like a dynamic *RecordSet*, but you cannot see records that other users add or delete. Data changes made by other users are still visible.

3. *Static*: The only type of *RecordSet* allowed when opening a client-side *RecordSet* object.

4. *Forward-only*: Behaves identically to a dynamic *RecordSet*, except that it allows you to only scroll forward through a *RecordSet*. A forward-only cursor is created by default.

Once you have a *RecordSet* with data, it is extremely easy to navigate through each of the fields. To demonstrate how this is done, we have provided the source code for a short ASP application. This program connects to the ACSA Questionnaire and prints a table with all of the database's values.

```
 1: <%@LANGUAGE="VBSCRIPT"%>
 2: <!--#include filel="ADOVBS.INC"-->
 3: <% Query = "SELECT Biographical_Data.*, Questions.Q1, Questions.Q2, "    &_
 4:        "Questions.Q3, Questions.Q4, Questions.Q5, Questions.Q6, "    &_
 5:        "Questions.Q7, Questions.Q8, Questions.Q9, Questions.Q10, "    &_
 6:        "Questions.Q11, Questions.Q12, Questions.Q13, Questions.Q14, " &_
 7:        "Questions.Q15, Questions.Q16, Questions.Q17, Questions.Q18, " &_
 8:        "Questions.Q19, Questions.Q20, Questions.Q21, Questions.Q22, " &_
 9:        "Questions.Q23, Questions.Q24, Questions.Q25, Questions.Q26, " &_
10:        "Questions.Q27, Questions.Q28, Questions.Q29, Questions.Q30, " &_
11:        "Questions.Q31, Questions.Q32, Questions.Q33, Questions.Q34, " &_
12:        "Questions.Q35, Questions.Q36, Questions.Q37, Questions.Q38, " &_
13:        "Questions.Q39, Questions.Q40, Questions.Q41, Questions.Q42, " &_
14:        "Questions.Q43, Questions.Q44, Questions.Q45, Questions.Q46, " &_
15:        "Questions.Q47, Questions.Q48, Questions.Q49, Questions.Q50, " &_
16:        "Questions.Q51, Questions.Q52, Questions.Q53, Questions.Q54, " &_
17:        "Questions.Q55, Questions.Q56, Questions.Q57, Questions.Q58, " &_
18:        "Questions.Q59, Questions.Q60, Questions.Q61 "               &_
19:        "FROM Biographical_Data INNER JOIN Questions ON "            &_
20:        "(Biographical_Data.Survey_Number = Questions.Survey_Number) AND " &_
21:        "(Biographical_Data.Squadron_Number = Questions.Squadron_Number) " &_
22:        "AND (Biographical_Data.Community = Questions.Community)"
23:  set connectobj = Server.CreateObject("ADODB.Connection")
24:  connectobj.Mode = adModeRead
25:  connectobj.Open "Survey2_DB", "sa", ""
26:  set records = Server.CreateObject("ADODB.RecordSet")
27:  set records = connectobj.Execute (Query, RecordsAffected, adCmdText)
28: %>
29: <html>
30: <head>
31: <title>The ACSA Questionnaire Database</title></head><body>
32: <table border="2" cellspacing="2" bordercolor="#800000">
33: <TR> <TD><font size="5"><b><i>Record Number</b></i></font></TD>
34: <%for each label in records.Fields%>
35:    <TD><font size="5"><b><i><%=label.Name%></b></i></font></TD>
36: <%next%>
37: </TR>
38: <% count = 1
39:   do until records.EOF%>
40:<TR><TD><font size="4"><%= count %></font></TD>
41: <%for each x in records.Fields%>
42:    <TD><font size="4"><%=x.Value%></font></TD>
43: <%next
44:  records.MoveNext
45:  count = count + 1 %>
46: </TR>
47: <%Loop
48: records.close%>
49: </table>
50: </body>
51: </html>
```

Figure 4.3. An ASP application (*test.asp*) that uses ADO to interact with the ACSA Questionnaire database.

So what does the code in Figure 4.3 do? Let's start with line 2. Here, the ASP application informs the web server to include the contents of another file before processing the contents of the current ASP file. The ability to use information from another file makes ASP even more flexible because it allows programmers to develop independent, reusable modules that can be shared among many ASP applications.

The command on line 2 is called a *"server-side include,"* and the net effect it has is identical to a C++ *#include* or Java™ *import* directive. (*Note:* the file *ADOVBS.INC* contains constant values used for database operations. For example, *adModeRead* has a value of 1, and is used in line 24 to indicate that the user is establishing a read-only connection. Because it is read-only, *INSERT, DELETE,* or *UPDATE* operations are not be allowed).

Lines 3 through 22 represent the SQL syntax to query the database. We are asking for all data values from both the *Biographical_Data* and *Questions* tables, and we are joining these tables together by their common attributes (i.e. the composite key fields *Survey_Number, Squadron_Number,* and *Community*).

When you finally reach Line 23, you come to the *Connection* object, which we create here and appropriately name *connectobj*. Two lines later, we indicate the data source we want to connect to - *"Survey2_DB"* - and include the username "sa" and password "" as parameters to the *Open* method. If all has gone well, we should now have a connection to our SQL 6.5 database and can use a *RecordSet* object to query it.

Line 26 illustrates the required syntax to create the *RecordSet* object. You will notice that we did not specify what type of cursor we are using. When we neglect to inform ADO what cursor type we want, the default is to create a *forward-only* one. Finally, on the very next line, we go through the *Connection* object to execute our query, and the results of this query are assigned to the variable *records*.

The ASP application then outputs regular HTML code until it comes to line 34, where we use a combination of logical VBScript statements to step through each row of the *RecordSet*. Lines 34 and 35 simply examine what the field names are for each row, and output these values. This is the very first row of the table that we are creating, and it tells the client what the data represents.

Because we want to keep a count of all records in the *RecordSet*, we initialize the *count* variable to 1 in line 38. Any time you execute a query (as we have done), the *RecordSet*, or cursor, is positioned at the very first tuple (if the data exists). This is unlike the JDBC model, where the cursor is placed prior to the first record and you must use the *ResultSet* method *next* to properly position the cursor before retrieving results. Since ADO does this for us, we can simply use a conditional statement to navigate through the *RecordSet* until we reach its end. The method *EOF* returns a Boolean indicating whether we have reached this point.

We are now ready to step through each field and output its value into a row of the table. This is done in lines 41 to 43, and when this step is completed, we reach line 44 which instructs the *RecordSet* object to move to the next tuple. We then increment the *count* variable and loop back to line 39 where we test to see if we are at the end of the *RecordSet*. This iterative process continues until we have spanned the entire *RecordSet*, and the output of this ASP application is a nice, tabular representation of the ACSA Questionnaire database as shown in Figure 4.4.

Figure 4.4. The output of the ASP application *test.asp*.

CursorType	Defines the type of cursor to be used with this *RecordSet*; must be defined prior to opening a *RecordSet*.
Open	Allows you to open a *RecordSet*. This method is overloaded so you can open a *RecordSet* without a *Connection* object (ADO still creates a *Connection* object, but it does not assign that object to a variable)

BOF	Boolean used to determine if you are at the beginning of a *RecordSet*.
EOF	Boolean used to determine if you are at the end of a *RecordSet*.
Move MoveFirst MoveLast MoveNext MovePrevious	Methods allowing you to navigate through a *RecordSet*.
AddNew Update Delete	Methods allowing you to add, update, or delete records from a database.

Table 4.8. Methods and properties of the *RecordSet* object.

3.   *COMMAND* Object

The *Command* object offers an alternative way of constructing and creating instances of *RecordSet* objects. It represents a command to be processed by the data provider, but since not all data providers have the capability to process commands, this object is optional [Ref. 9]. You should use this object for storing procedures (creating a group of SQL statements that perform a logical task), executing bulk operations, or manipulating the structure (schema) of a database.

The Command object is composed of Parameter objects, which represent the parameters associated with queries or stored procedures. You might choose to use this object when you are continually executing queries, many of which only differ by the WHERE or ORDER BY criteria at the end of the SQL statement. By creating a stored procedure, you are telling the DBMS to keep a set of SQL statements that you provide handy for future use. So when you want to perform this query later, the DBMS does it for you, using any parameters you pass to the procedure. This process greatly enhances database performance (Note: JDBC used the *CallableStatement* for the same purpose). Table 4.9 lists the more commonly used methods and properties of the *Command* object. Because we do not use the Command object in our ASP application, use and syntax of this object should be referenced from another source such as the ADO API documentation.

CommandText	Defines the executable text of the command, e.g., a SQL statement.
CommandTimeout	Sets the amount of time a data provider will execute a command.

Table 4.9. *Command* object properties.

4.   *ERRORS* Object

Any operation you perform with ADO has the potential to generate one or more errors. These errors represent specific data provider errors, not ADO errors. If a problem occurs and you need to know what happened, you can reference the *Errors* collection, which is composed of these objects. This collection is cleared every time a new error occurs.

The *Errors* object, as you might expect, contains only properties that you can investigate to determine the nature of the problem:

- The *Description* property contains the text of the error.

- Every error has a number associated with it. You can determine what error occurred by looking at the *Number* property, and if you choose, have your ASP application perform some action based on the error code.

- The *Source* property identifies the object that raised the error.

- *SQLState* and *NativeError* properties provide information from SQL data sources, e.g., primary key violations, fields cannot have null values, etc…

D.   ACSA ASP

Putting it all together requires a solid knowledge of the ASP and ADO object model. With that hurdle now cleared, we can discuss how ASP is used in the ACSA web site.

For this thesis, we developed two ASP applications which are used to access the ACSA databases. One database allows the client to interact with the *Num_List_DB* database, which contains a listing of valid survey numbers for every squadron participating in the survey, and the other lets the client view survey submissions from clients. The source code for each ASP application is almost identical, with small variations that correspond to the field names of the database tables. Each application has three files that were auto-generated using Microsoft's Visual InterDev.

Because the code is extremely similar, we will pay meticulous attention to the portion that allows an authorized administrator to interact with the *Numbers* database. Understanding how similar code works with the *Survey2_DB* database is left as an exercise for the

reader. You can find all seven files – the *global.asa* file plus three files for each ASP application – in Appendix H.

The shortest and by far simplest file to examine is *global.asa*:

```
1: <SCRIPT LANGUAGE="VBScript" RUNAT="Server">
2: </SCRIPT>
3: <SCRIPT LANGUAGE=VBScript RUNAT=Server>
4: Sub Session_OnStart
5:          Session("Num_List_DB_ConnectionString") = "DSN=Num_List;Description=squadron
6: survey access listing;SERVER=SPITFIRE;UID=sa;PWD=;APP=Microsoft (R) Developer
7: Studio;WSID=SPITFIRE;DATABASE=Num_List_DB"
8:          Session("Num_List_DB_ConnectionTimeout") = 15
9:          Session("Num_List_DB_CommandTimeout") = 30
10:         Session("Num_List_DB_RuntimeUserName") = "sa"
11:         Session("Num_List_DB_RuntimePassword") = ""
12:         Session("Survey2_DB_ConnectionString") = "DSN=Survey2_DB;Description=Survey
13: Database;SERVER=SPITFIRE;UID=sa;PWD=;APP=Microsoft (R) Developer
14: Studio;WSID=SPITFIRE;DATABASE=Survey2_DB"
15:         Session("Survey2_DB_ConnectionTimeout") = 15
16:         Session("Survey2_DB_CommandTimeout") = 30
17:         Session("Survey2_DB_RuntimeUserName") = "sa"
18:         Session("Survey2_DB_RuntimePassword") = ""
19: End Sub
20: </SCRIPT>
```

Figure 4.5. The file *global.asa*.

When a user requests either ASP database application, an *Application* object and then a *Session* object are created. The subroutine called *Session_OnStart* is processed beginning at line 4. Lines 5 through 12 define several session variables that are referenced later in the ASP application code. This information includes the data source name of either database, how long the web server will wait to make a connection or execute a command before terminating the attempt (lines 8, 9, 15 and 16), and predefined client usernames and passwords (lines 10, 11, 17, and 18). Because these variables persist throughout the lifetime of the session, we can conveniently access them when needed.

Once the web server is done processing this file, the client is directed to a tabular listing of the database he requested to see (see Figure 4.6).
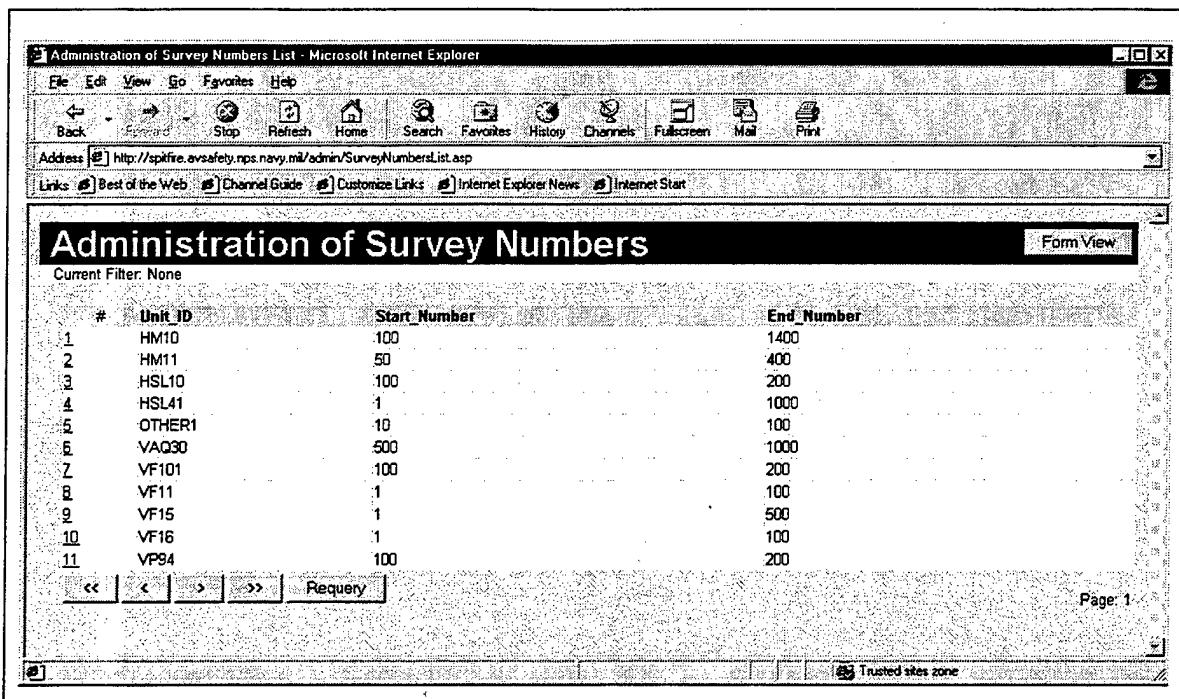
Figure 4.6. ASP application displaying the contents of the *Numbers* database.

The ASP file that processes this database view is *SurveyNumbersList.asp*. This 454-line file contains extremely detailed source code, providing the client with the ability to navigate through all records contained in the database. We will take care to explain portions of this file that we feel are worth knowing about. More importantly, however, you should have a good idea of the manner in which these files interact with one another. For example, clicking on the "Form View" button as shown in Figure 4.6 actually sends a command to the *SurveyNumbersForm.asp* application. In this case, the *Request* object that is sent to this application contains two parameters:

1. a hidden attribute called *FormMode* with a value set to *Edit,* and

2. a visible *Submit* button that is named *DataAction* and has the value *"Form View"* (this is the button you see in the upper right-hand corner of Figure 4.6).

The *SurveyNumbersForm.asp* application contains VBScript code that evaluates the parameters that are passed to it:

```
1: <%
2: strFormMode = "Edit"    ' Initalize the default
3: If Not IsEmpty(Request("FormMode")) Then strFormMode = Request("FormMode")
4: If Not IsEmpty(Request("rsSurveyNumbersSQLQuery_PagingMove")) Then
5:    StrPagingMove = Trim(Request("rsSurveyNumbersSQLQuery_PagingMove"))
6: End If
7: %>
```

Figure 4.7. Evaluating the Request object parameters passed by
SurveyNumbersList.asp.

The variable *strFormMode* is set to a default value of *"Edit"* in
Line 2. The next line determines if the *FormMode* parameter has a value.
If the parameter has no value, then the if block is exited, otherwise,
we assign the value of this paramter to the variable *strFormMode*.
We know that the *FormMode* paramter does indeed have a value, namely
"Edit", so this value is assigned to *strFormMode*. Lines 4 and 5 are
disregarded, because a value for the variable
rsSurveyNumbersSQLQuery_PagingMove was never sent. The *SurveyNumberForm*
application then outputs HTML code that generates a "form" view of the
data (see Figure 4.8).
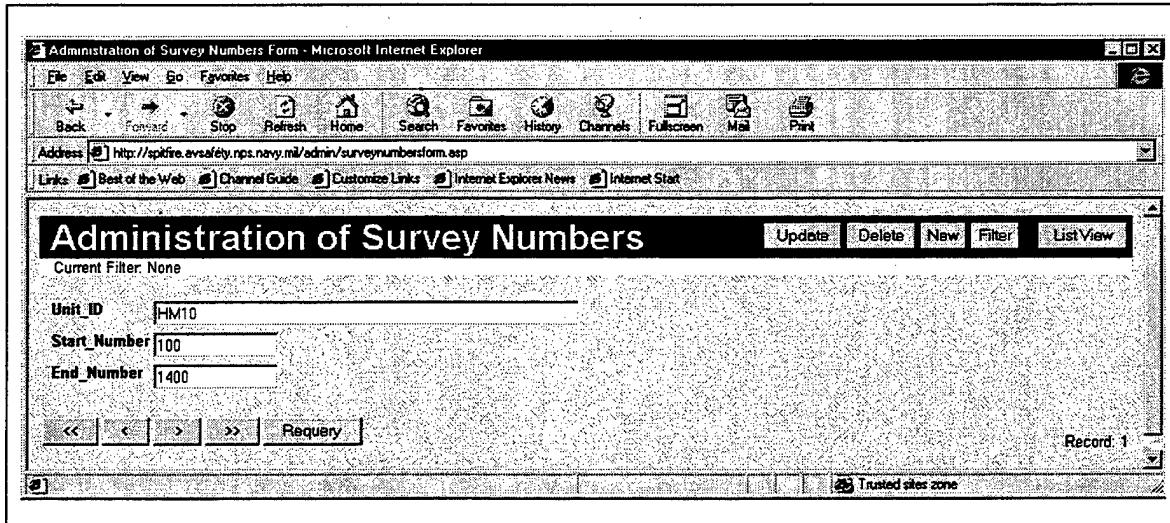
Figure 4.8. The "form" view of the *Numbers* database (file
SurveyNumbersForm.asp).

From here, the client can update, delete, filter or navigate (via
the "<<", "<", ">", or ">>" buttons at the bottom of the of the page)
through records, and go back to the list view that was originally
displayed. Each of the buttons on the top-right of the page are
assigned the name *DataAction,* and their values are set to the names
that appear on them. When you click on one of these buttons, another

71

Request object is created, parameters are given values, and the data is sent to the third file *SurveyNumbersAction.asp*. This can be viewed as an intermediary ASP application, which determines and processes your request. This file is also quite long - 725 lines - but includes the code required to tie the applications together. Clicking on any of the buttons in the *SurveyNumbersForm.asp* application takes you to the event handler shown in Figure 4.9.

```
1: <%
2: If Not IsEmpty(Request("DataAction")) Then
3:       strDataAction = Trim(Request("DataAction"))
4: Else
5:       Response.Redirect "SurveyNumbersForm.asp?FormMode=Edit"
6: End If
7: '-----------------
8: ' Action handler
9: '-----------------
10: Select Case strDataAction
11:       Case "List View"
12:             Response.Redirect "SurveyNumbersList.asp"
13:       Case "Cancel"
14:             Response.Redirect "SurveyNumbersForm.asp?FormMode=Edit"
15:       Case "Filter"
                  ...
16:       Case "New"
                  ...
17:       Case "Find"
                  ...
18:       Case "All Records"
                  ...
19:       Case "Apply"
                  ...
20:       Case "Insert"
                  ...
21:       Case "Update"
                  ...
22:       Case "Delete"
                  ...
23: End Select
24: %>
```

Figure 4.9. Processing application requests (Note: "..." denotes omitted code).

This excerpt from *SurveyNumbersAction.asp* first sets the variable *strDataAction* to the value that was set in the *Request* object. The *Select* structure starting on line 10 determines what button the user clicked, and executes code (omitted for clarity) that handles the given event. In the simplest case, i.e., we click the *"List View"* button to return to

72

the tabular representation, the client is redirected to the
SurveyNumbersList.asp file (line 12).
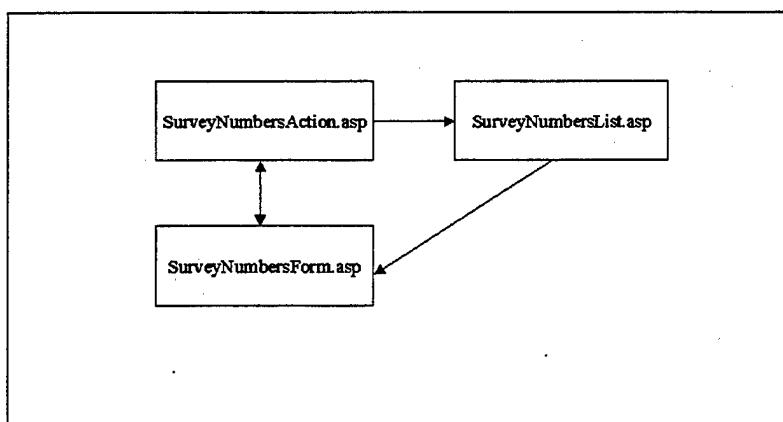
Figure 4.10. ASP application interaction.

As previously mentioned, we used Microsoft's Visual InterDev to
develop these ASP applications. The files for the *Survey2_DB* database
operate in the exact same manner as the files for the *Num_List_DB*
database. Visual InterDev provided all the code that was required to
navigate through recordsets and perform operations on the data source.
While this software product made web-driven database applications
rudimentary to develop, we did encounter an application error when
trying to generate the ASP files for the *Survey2_DB* database. As a
result, we were forced to manually edit code for this application.
Doing so required a good understanding of how the ASP application
worked. This problem is most likely due to the large size of the
Survey2_DB database, as we have not encountered a similar error when
working with smaller, more manageable databases.

One last point worth mentioning is that if the database schema is
ever changed, then the ASP application will have to be regenerated and
redeployed. The same limitation holds true for the Java™ applets and
HTML/servlet interface. Creating new ASP applications to allow clients
to input and retrieve data will be far easier than redoing the other
ACSA components. For this reason, we highly encourage the reader to
become familiar with how to use the Visual InterDev product.
Reference 9 provides an outstanding discussion on this subject.

E.   CONCLUDING REMARKS

ADO is as easy to use and learn as ASP, and with a tool like Visual InterDev, you can build ASP applications in little time that allow you a great deal of flexibility. ADO is inexpensive (free), and like Sun Microsystem's "*sun.jdbc.odbc.JdbcOdbcDriver*," allows you to communicate with ODBC data sources. While the *JdbcOdbc* driver from Sun Microsystems uses a non-standardized object model, ADO has a well-defined API. Another factor to consider when deciding which object model to use for database programming is that ADO enhancements will allow you to communicate with non-SQL data sources. JDBC™ does not offer this capability, nor are there any plans to expand its features to allow this. We have included Table 4.10, which lists the major JDBC™ objects and their ADO equivalents, for your reference.

JDBC™ object	ADO equivalent
DriverManager	*Connection*
Connection	*Connection*
Statement	*Command*
PreparedStatement	*Prepared Command*
CallableStatement	*Command*
ResultSet	*RecordSet*

Table 4.10. JDBC™ objects and their ADO equivalents [Ref. 24].

V. ACSA SYSTEM DEVELOPMENT AND OPERATION

A.   INTRODUCTION

This chapter contains a great deal of information that is
pertinent to ACSA system users, administrators, and future project
developers. It serves not only as a system user manual, but also as a
guide through the development decisions the original authors made. You
should read this chapter until you are familiar with its contents in
their entirety. We will cover alot of ground here - from the design of
the two ACSA databases and the use of the Microsoft Access upsizing
utility, to how we used Microsoft's NT Server 4.0 and the company's
BackOffice suite of products to support this project.

B.   DATABASE DESIGN

The ACSA project uses two different SQL 6.5 databases. The
Survey2_DB database contains client survey submissions, while the
Num_List_DB is a utility database referenced by the servlets and applets
to determine if the client is an authorized survey participant. Both
databases were initially developed in Microsoft Access, and
subsequently upsized to Microsoft's SQL 6.5 database format. Access is
a powerful, easy-to-use Relational Database Management System (RDBMS),
but it is limited in its application as a multi-tier database provider.

As our research efforts progressed, we found a need for a more
robust DBMS that could reliably respond to client requests. The two
major products available at the time were Microsoft's and Sybase's SQL
Server. We chose to work with the Microsoft product for a number of
reasons, to include:

- Microsoft provided a MSAccess add-in utility that allowed you
  to easily upsize to a SQL 6.5 database without losing the
  database schema or its contents.

- The United States Navy had promulgated the IT-21 policy. This
  policy emphasized the need for developers to use Microsoft
  technology in their projects.

Microsoft's SQL Server is part of Microsoft's BackOffice suite of
applications designed to run exclusively on the Windows NT Server
operating system. You can interact with any database by using
Structured Query Language (SQL) syntax, which is what you have seen our
programs do in the previous chapters. These programs can communicate
with any database managed by SQL Server, i.e., think of SQL Server as a

container that holds many databases (the *Survey2_DB* and *Num_List_DB* are just two of the many that it manages). When a program sends a request to SQL Server, SQL Server takes that request, finds the database that the program wants to interact with, and then manages the interaction with that database and returning the information to the client (or program) that requested it.

In order for us to communicate with SQL Server, there were a number of steps we had to take, to include properly upsizing the original databases (this process is described shortly), and using a driver from our Internet applications that was capable of taking the SQL calls we made and communicating with the SQL Server process running on the *spitfire* machine. We decided to use WebLogic's *FastForward* driver (see Chapter II for additional information on this product) for this purpose.

Perhaps the best place to begin discussing how the system was designed is to look carefully at the schema (or design) of both databases.

1. **The *Survey2_DB* database schema**: We designed a relational schema for the *Survey2_DB* database patterned after the existing survey form. This form contains both biographical information and survey questions that the participant must complete before the survey is submitted for further processing. The logical separation we made was to create two database tables - one that would hold the client's biographical information, and the other which would contain the respondent's selections to the 61 survey questions. The attributes and data types for the *Biographical_Data* and *Questions* tables can be found in Tables 5.1 and 5.2 below.

A composite primary key is used to uniquely identify each record in either table. This key is composed of three fields: *community, Squadron_Number* and *Survey_Number*. With this key, we can link the records in both tables together, and for any record, we can identify all information that the client submitted. It should be noted that at no time can the identity of the client be determined. The granularity of such an inquiry is limited to the squadron the submission was received from, and this information is only available to ACSA system administrators. If this implementation does not afford enough protection, squadrons can use an "Other" category to identify themselves.

Without some way of curtailing how many surveys a client can submit, it is possible that the survey results can become skewed in such a manner that the whole survey process becomes analytically worthless. In order to preclude this possibility from occurring, the ACSA program administrator must authorize a block of unique survey numbers for each squadron (these numbers are found in the *Num_List_DB* database). These blocks of numbers should be randomly generated to prevent the possibility that one survey participant will guess what another squadron's block of valid survey numbers are.

FIELD	DATA TYPE	POSSIBLE VALUES
Community	text	*Primary key field* - community. Possible values are HC, HCS, HM, HMH, HMT, HS, HCS, HSL, VAQ, VAW, VF, VFA, VMAQ, VP, VQ, VR, VRC, VX, VT, VX, OTHER
Squadron_Number	number	*Primary key field* - squadron number
Survey_Number	number	*Primary key field* - authorized survey submission number
rank	number	Possible values are 0 to 3: 0 = O1-O3, 1 = O4-O6, 2 = E1-E5, 3 = E6-E9
Designator	text	Possible values are 0 to 2: 0 = Pilot, 1 = NFO, 2 = Aircrew
Flight_hours	number	Number less than 25,000
Type_hours	number	Number less than flight_hours entry
Dept_head	text	Either YES or NO
Status	text	Either Regular, Active Reserve or Drilling Reserve
Service	text	Possible values are 0 to 2: 0 = USN, 1 = USMC, 2 = Other
Location	text	Either East Coast, West Coast or Other

Table 5.1. *Survey2_DB Biographical_Data* table schema.

FIELD	DATA TYPE	POSSIBLE VALUES
Community	text	*Primary key field* - community. Possible values are HC, HCS, HM, HMH, HMT, HS, HCS, HSL, VAQ, VAW, VF, VFA, VMAQ, VP, VQ, VR, VRC, VX, VT, VX, OTHER
Squadron_number	number	*Primary key field* - squadron number
Survey_number	number	*Primary key field* - authorized survey submission number
Question 1 ... 61	number	Array index value: 0 = Select a Question 1 = Strongly Disagree 2 = Moderately Disagree 3 = Slightly Disagree 4 = NEUTRAL 5 = Slightly Agree 6 = Moderately Agree 7 = Strongly Agree 8 = NOT APPLICABLE

Table 5.2. *Survey2_DB Questions* table schema.

2. **The *Num_List_DB* database schema:** Before a client can submit a completed survey to the database, they must be assigned a unique survey number. This survey number is one of a block of

numbers that the administrator has assigned to the squadron in
the *Num_List_DB*. This is, to a large part, the only
administrative overhead that must be incurred before the
survey can be appropriately administered to fleet squadrons.
It is important that these authorization numbers are passed
only to the local squadron ACSA survey facilitators. The
schema for this *Num_List_DB* database is shown in Table 5.3.

FIELD	DATA TYPE	VALUE
Unit_ID	Text	*Primary key*, individual squadron number
Start_Number	Number	First number in a series of authorized survey submission numbers
End_Number	Number	Last number in a series of authorized survey submission numbers

Table 5.3. The *Num_List_DB* table schema.

Other database design considerations that future developers
should be aware of include the following:

- If the ACSA program administrator decides to change the
  wording of questions, the code for all project components will
  need to be modified. This could have been precluded if the
  questions were imported from an external file (such as a text
  file), so future developers should seriously consider
  separating the questions from the ACSA programs. A change in
  the way a question is worded does not impact the SQL 6.5
  databases.

- If the number of questions change, developers will have to
  modify the back-end SQL 6.5 *Survey2_DB* schema, as well as the
  ACSA program code. Although this may be a somewhat tedious
  task, it should not be too time-consuming of a process to
  complete. The difficult part will be if the number of
  questions expands beyond the current 61, but we do not
  anticipate this happening.

- Adding, deleting, or renaming selections for currently
  existing fields should be neither difficult nor time-consuming
  to do. Upon the conclusion of the first usability testing, we
  made several changes to the rank field of the HTML, ASP, and
  applet front-ends that took less than ten minutes to complete.
  We did not have to modify the SQL 6.5 databases.

Clients who are survey participants can query the contents of the

Survey2_DB database but cannot directly perform *UPDATE* or *DELETE*

operations. Once records are entered into the database, the ACSA

program administrator, who retains full control over the database, is

the only person that can change them. If changes need to be made, the

ACSA administrator can make them from the portion of the web site that

allows administrative access, or by going to the *spitfire* machine and

directly accessing the data through the MS Access front-end.

A final requirement that we had to satisfy was protecting the identify of the individual squadron. This was important for our development of the *Results* applet, because we did not want Commanding Officers to be able to continuously monitor the results for their squadrons, or to compare their squadron's results to others. Therefore, the granularity of queries is limited to a particular community. We recognize that feedback is an important element of any survey process, and so it was decided that Commanding Officers would be able to obtain their squadron results, but only through separate communication to the ACSA program administrator requesting this information. Future upgrades to the ACSA project could include automating the process whereby Commanding Officers can receive results for their squadrons instantly.

Database disk storage requirements should be minimal. Based on previous surveys, we estimate a less than 3MB storage requirement for each annual survey but have allocated 50MB of disk space to comfortably accommodate up to 16 annual surveys. This container can be resized at any time by using the SQL Enterprise Manager application.

C.   THE SQL SERVER 6.5 DBMS

Since the two databases were initially designed in Microsoft Access, we needed some convenient means of transferring the schema and current data to SQL 6.5 format. If no such utility existed, we would have to recreate the tables in each database from scratch, and we would have to manually input new data to continue testing the prototype system. Fortunately, Microsoft provided an Access upsizing utility add-in, which could automatically take your Access database and port it over so you could use it with SQL 6.5. Meanwhile, you could use Access as a front-end to view the upsized SQL 6.5 database or you could use another add-in that was installed at the same time the original upsizing tool was added. This utility, the SQL-Server Browser, can be selected by going to *Tools->Add-Ins* from the MSAccess menu.

Figure 5.1 shows what happens when you open the *Survey2_DB* in Access. The SQL Server tables are represented with a small arrow and globe icons preceding the table name. The upsizing utility automatically creates these new SQL Server tables, either with or without the associated table data. Additionally, the previous Access tables are renamed by appending a *_local* label to the end of the table name. Program administrators should be very careful when using this

interface to change the contents of the *Survey2_DB*. If you want to change the actively used ACSA databases, ensure that you are using the SQL 6.5 tables denoted with the arrow and globe icons. Changes to the Access tables will have no impact on the ACSA project.
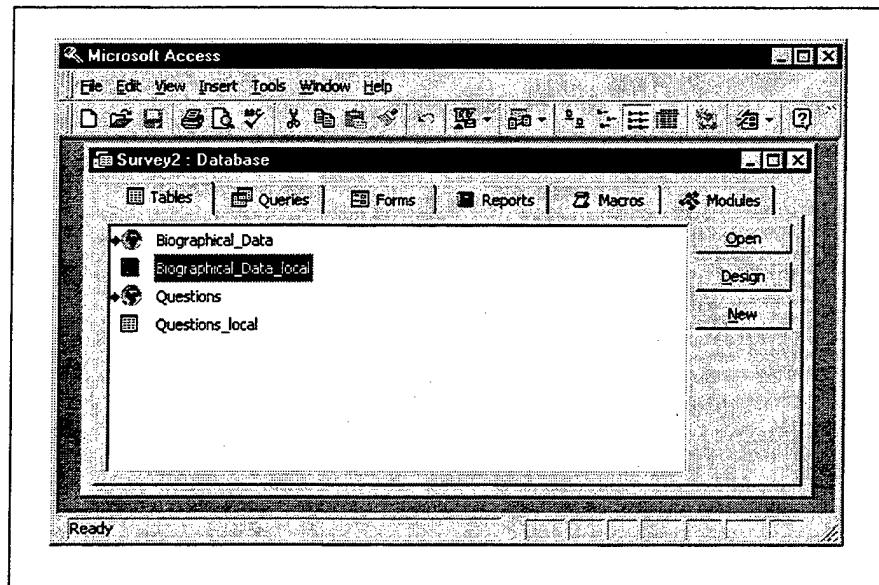
Figure 5.1. SQL 6.5 *Survey2_DB* database tables as seen from MSAccess.

The SQL Enterprise Manager application is the main interface between SQL server and the databases it manages. You can find this application by selecting *Start-> Programs-> Microsoft SQL Server 6.5-> SQL Enterprise Manager*. The Enterprise Manager is used to manage and configure SQL servers, monitor server activity, create database devices, and creating new databases. A representative view of what the Enterprise Manager shows you is illustrated in Figure 5.2. In the remainder of this section, we will briefly discuss the ACSA components that can be found in the Enterprise Manager on the *spitfire* computer.

Microsoft SQL Enterprise Manager

File   View   Server   Tools   Manage   Object   Window   Help

Server Manager

Server: SPITFIRE

- SQL 6.5
  - SPITFIRE (SQL Server 6.50)
    - SQL Mail
    - SQL Executive
    - Distributed Transaction Coordinator
    - Database Devices
      - FortuneDB_Device
      - master
      - MSDBData
      - MSDBLog
      - Num_List_Device
      - Survey2_Device
    - Backup Devices
    - Databases
      - Fortune_DB
      - master
      - model
      - msdb
      - Num_List_DB
      - pubs
      - Survey2_DB
        - Groups/Users
          - public
        - Objects
          - Tables
            - Biographical_Data (dbo)
            - dtproperties (dbo)
            - Questions (dbo)
          - Views
          - Stored Procedures
          - Rules
          - Defaults
          - User Defined Datatypes
      - tempdb
    - Logins

Ready   \SPITFIRE\

Figure 5.2. The *spitfire* Enterprise Manager window.

Things to take note of from Figure 5.2 are:

- The traffic light icon is currently green, indicating that the
  server is up and running. You can use the SQL Service Manager
  program (select *Start->Programs->Microsoft SQL Server 6.5->SQL Service
  Manager*) to pause, stop, restart, or resume the SQL Server
  service. You can likewise accomplish the same thing by right
  clicking on the icon and selecting the appropriate option. If
  clients notice problems trying to submit their surveys, you
  may want to stop and restart the SQL Server service.

81

- Under the *Database Devices* folder, you will notice several devices. These are essentially containers that are setup prior to the actual database to indicate how much storage space should be used for data and log files. Only the *Num_List_Device* and *Survey2_Device* are ACSA program devices.

- The *Databases* folder contains the actual databases used by a device. As you can see, there has to be an association between a device and a database, and this association is maintained, managed, and monitored by the SQL Server application.

- In the *Objects* folder of each database, you will find a list of tables, views, stored procedures, rules, defaults, and user-defined datatypes. Opening the *Tables* folder of the *Survey2_DB* database reveals the tables you might expect to find there - *Biographical_Data* and *Questions*. From Enterprise Manager, you can only access the table schema and not the actual table data.

In order to understand how a database device is properly created, let us take some time to review this process.

Before upsizing, you need to create a SQL Server *database device* to hold the database. This database device is a container that has a space reserved for data and for logging interactions with the database. To create a database device from the Enterprise Manager, you can either right click the *Database Devices* folder or select *Manager-> Database Devices-> New Device* from the menu. This will open a dialog box and prompt you for the information necessary for creating a new device. The *Survey2_DB* database device was created with the settings listed in Table 5.4.

ENTRY	OUR SELECTION	PURPOSE
Name	Survey2_DB	Database file name
Location	F:Survey2_Device.DAT	Database file location
Size	50MB	Initial database size; this value can be changed later.

Table 5.4. *Survey2_DB* database device settings.

D.   UPSIZING THE SURVEY2_DB ACCESS DATABASE TO SQL SERVER

The upsizing utility is an Access add-in program developed after the release of Access 97. This freeware utility can be downloaded from Microsoft's web site at [http://www.microsoft.com/]. This section will lead the reader through the step-by-step process that we used to upsize our Access *Survey2_DB* database to SQL Server. If you would like additional information, we recommend taking a look at the Platinum Edition Using Access 97 by Roger Jennings [Ref. 25].

Prior to installing the upsizing utility, ensure that MSAccess is not running. After you have successfully run the upsizing utility

program (*AUT97.exe*), open your *.mdb* Access database and select *Tools->Add-ins->Upsize to SQL-Server*. This selection will initiate the upsizing wizard that is designed to guide users through the upsizing process. Since we are upsizing an existing database, *Survey2_DB*, select the radio button *Use existing database,* then click *Next.* Make the following selections:

- Select the *Machine Data Source* tab, and then click *New*.

- Select *System Data Source (Applies to this machine only)* then click *Next*.

- Highlight *SQL Server,* and then click *Finish.*

This action will bring you to the first of five screens for creating a new SQL Server data source. At each screen, we made our selections as follows:

- Wizard Screen 1: We entered *Survey2_DB* as the data source name, our data source description (optional) and then select *local* server, then *Next.*

- Wizard Screen 2: Select *With Windows NT authentication using the network login ID* and *Connect to SQL Server to obtain default settings for the additional configuration options* check box, then *Next.*

- Wizard Screen 3: Enter *Survey2_DB* in *Change the default database to* text box. Then select *Create temporary stored procedures for prepared SQL statements and drop the stored procedures* check box, select *Only when you disconnect, Use ANSI quoted identifiers,* and *Use ANSI nulls, padding, and warnings* check boxes, then *Next.*

- Wizard Screen 4: Select *Let SQL Server ODBC driver choose the translation method,* then *Next.*

- Wizard Screen 5: select *finish*.

E.   WEB SERVERS

In order to serve files to web clients (browsers), you need a network application that can intelligently interpret and fulfill client requests. The application that performs this service is called a web server. Web servers are background processes that run transparently to system users. This means that if you are working on a computer that has a web server installed and running, you do not have to wait for the server to complete before executing additional tasks or applications (i.e., the web server does not impact how you go about your other business). This ability - to run multiple processes simultaneously on a

single computer - is known as multitasking, a feature of modern computer architecture that greatly enhances the utility and functionality of computer systems.

Without multitasking, web servers and many other present-day applications would be extremely burdensome to use. For instance, web servers could only process one client request at a time and input and output (I/O) operations (such as writing to a log file) would seriously degrade application performance. Multitasking introduced the concept of multithreading, which is the ability of a program to have multiple parts that are executing simultaneously. Web servers could then divide labor into modules, starting a new thread for each client requests and other threads for input and output operations. By breaking the application into multiple parts that can execute in parallel, the operating system can rapidly switch between these threads and divide processor time between them. In this manner, web servers become more responsive, run smoother, and clients' requests aren't left sitting in a long queue for processing.

Whatever web server you decide to use, it should be built in such a way that it takes advantage of the benefits offered by multithreading. We experimented with several freeware web servers, to include Apache, OmniHTTPd, ServerSeven, and Internet Information Server (*IIS*) 4.0. While all of these products were able to handle multiple, simultaneous client requests in an efficient manner, the former three could not take advantage of the Windows NT Security Accounts Manager (SAM) database, process Active Server Pages (ASP), or serve files that had a "$" in them. This last restriction posed a serious problem when the client requested any of the Java™ applets, because many of the compiled class files that were part of the applet had "$"'s in them.

For these reasons, we decided to use and experiment with Microsoft's IIS. How we configured IIS for the ACSA project is the subject of this chapter. By the time we are done exploring this subject, the reader should have a good understanding of how IIS is configured, what components work in conjunction with IIS and how they are used, how to take advantage of Windows NT security features, and knowledge of good security practices that can make web sites more secure (not invulnerable) from unwanted intruders.

84

F. WINDOWS NT 4.0 SECURITY

IIS is tightly integrated with Windows NT and can take advantage of the security features native to an NT machine. But before we examine how IIS security features are enabled, it is important that we review how NT security works in general, particularly if this is the first time that you have been exposed to the NT operating system.

Everything done in NT revolves around the use of what is known as a "user account." When you first log on to an NT machine, you are presented with a splash screen that asks for your username and password. After you enter this information, the *WINLOGON* process (a program running on the NT machine) accesses the Security Accounts Manager (SAM) database and determines whether the information supplied is valid (note that the *NETLOGON* process is responsible for this task when shares are created and network users want to access these shares).

You should remember that supplying a username and password alone does not confirm whether the user is legitimate or not. It could be that both the username and password were compromised and are now being used by someone who is not authorized access to the system. To ensure that users are indeed legitimate, if this is a stringent requirement that you want to impose on system users, you should use a supplementary biometric device such as a fingerprint scanner.

If the username and password correspond to an entry in the SAM, then a security token is generated. This token is a "key" that will either allow or disallow the user access to portions of the system that reside on a New Technology File System (NTFS) partition. NTFS is a special file system used by NT that uses Access Control Lists (ACLs) to enumerate what users or groups have access to files and directories, and what their rights are to these objects.

File system security is lost without NTFS. In fact, if you install the NT operating system on a File Allocation Table (FAT) partition, you will notice that when you look at the properties of a file or directory in a utility such as Windows NT Explorer, you don't have the ability to set security permissions. Hence, if you have a requirement to set file-level permissions, you should install NT on an NTFS partition. You can always change a FAT partition to a NTFS partition (but not vice versa) by using the dos *CONVERT* utility. For

more information on how this program works, type *"convert/?"* at the dos command prompt.

So you might be wondering how user accounts and file system security are related. An example will best illustrate how accounts are created, managed, and the relationship between accounts and file system security. First, let's look at the *User Manager for Domains.*

This administrative tool (shown in Figure 5.3) is one of several provided by the Windows NT operating system. It gives users with administrative privileges the ability to manage the permissions of user and group accounts.

If you look at Figure 5.3, you will see that we have eleven accounts on our machine: *Administrator, Commander, DocC, dont, Guest, IUSR_SPITFIRE, IWAM_SPITFIRE, jsheld, mingo, Replicant,* and *SQLExecuteCmdExec.* The first two accounts are created by default whenever you install Windows NT. The *Administrator* account is the one you will use when you want to log on with administrator privileges. Since everyone who installs Windows NT knows that this account exists, it is highly recommended that you rename the account to something less obvious and use a password that is hard to guess. You can rename the account by clicking once on it and then selecting *Rename* from the *User* menu of the *User Manager for Domains* program. To change a password, double-click on the account and enter the password twice where prompted.

The *Guest* account has no password and is used to provide guest access to the computer, the domain, and the resources that are being shared. If Local Area Network (LAN) users do not explicitly have an account listed in *the User Manager for Domains,* then they log onto the machine with the permissions granted to the *Guest* account. Leaving this account enabled is extremely dangerous, because it allows anyone access to your system. Since all newly created shares give full control permission to everyone, including guests, security can quickly become a problem. For this reason, we highly recommend disabling this account (which was done on the *spitfire* machine hosting the ACSA project). If users still require access to domain shares, you should create an account for them using *User Manager for Domains.*

When you click twice on the *Guest* account, you open a window identical to that shown in Figure 5.4. This menu allows you to set many

user account properties, such as the groups that the user is a member of, what hours they can log on, if they are allowed to change their password, etc...
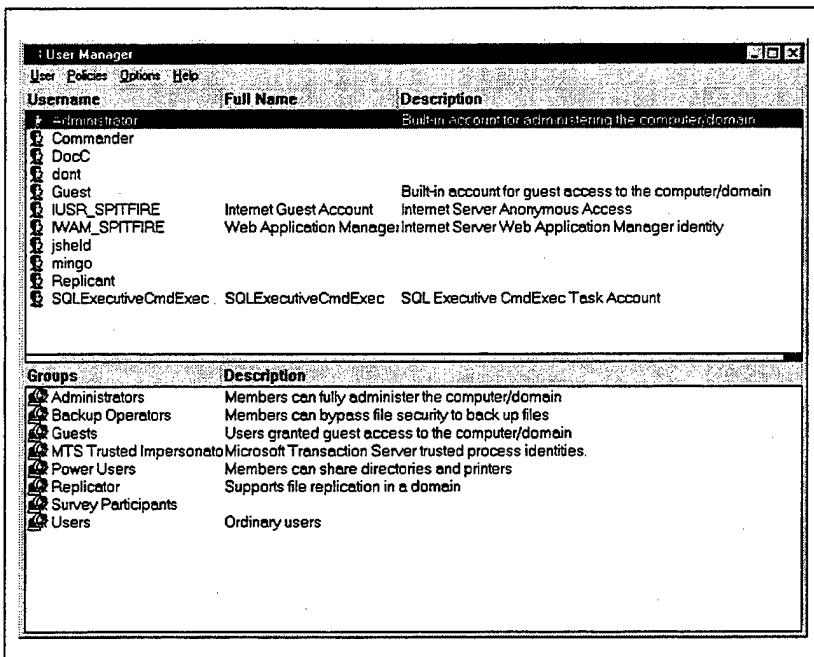
Figure 5.3. User Manager for Domains.

The accounts *Commander, DocC, dont, jsheld, mingo*, and *Replicant* are also user accounts. But what happens when a process wants its own account? As you might have already guessed, a process can also be associated with a user account. This is the case with IIS and SQL Server 6.5. IIS uses the account *IUSR_<MACHINE NAME>*, where the machine name is computer name specified in the identification tab of the control panel network applet, as the Internet guest account. *IUSR_SPITFIRE* is the Internet guest account in this example and is created, along with *IWAM_SPITFIRE* account when IIS is installed.

The *IUSR_SPITFIRE* account is accessed whenever a web client sends a request to your IIS web server. When this occurs, the client connects to your machine with the permissions granted to the IUSR_SPITFIRE account. Similarly, SQL Server 6.5 uses the *SQLExecutiveCmdExec* account when a client sends DBMS requests. You should **NEVER DELETE** these accounts. You should also know that you **CAN NEVER RECREATE** an account, even if you give the account the same access permissions as the original one. This is because each account has a Security IDentifier (SID) which is unique and never repeated.

87
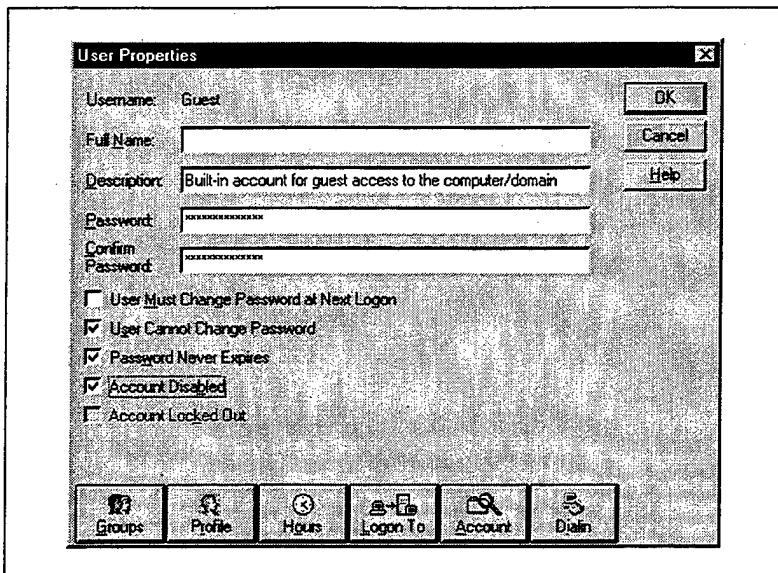
Figure 5.4. Changing user account properties.

When the number of domain users is small, managing permissions

through user accounts is easily done. But suppose we have 1,000 users.

We could then potentially find ourselves, as administrators,

spending most of our time changing file or directory-level permissions.

To circumvent this problem, NT provides groups. How groups help

administer security permissions is clearly seen if you think of a large

corporation that has many departments. As an administrator, we might

create a group for each department - one for MARKETING, FINANCE, SALES,

RESEARCH_AND_DEVELOPMENT, etc... Each group, in turn, has a list of

users that are members of that group. In this manner, we can set

permissions based on groups rather than individual users. If a user

doesn't have access to a resource that he needs, you can either add him

to a group that does have access, or explicitly allow that user access.

Group management eases the administrative burden of applying security

permissions.

The ACSA project has only one group - *Survey Participants.* The users

dont, DocC, and *Commander* are members of. This group has permission to

submit a survey using either the HTML form or the Java™ applet. It is

far easier to set directory permissions by group rather than by account, and you should realize that granting *Survey Participants* full control access to a directory's contents is in effect the same thing as granting the individual users *dont, DocC* and *Commander* full control.

With this setup, everyone can submit data to the *Survey2_DB* database managed by SQL Server. But we want to impose a restriction on who can view the data that has been submitted. The *Results* applet is the mechanism used to query and view database results, so we restrict access to the directory that contains the code for the *Results* applet to the *Commander* and *DocC* accounts. This means that clients who log on using the *dont* account will not be able to access this portion of the web site. Setting permissions on directories can be done either in IIS or through Windows NT Explorer.

In Explorer, simply left click on the directory, which will highlight it, then right click and select *Properties*. You should see a dialog box that has four tabs: *General, Web Sharing, Sharing,* and *Security*. If you move to the *Security* tab, and select *Permissions*, you should then see a *Directory Permissions* dialog box similar to Figure 5.5.
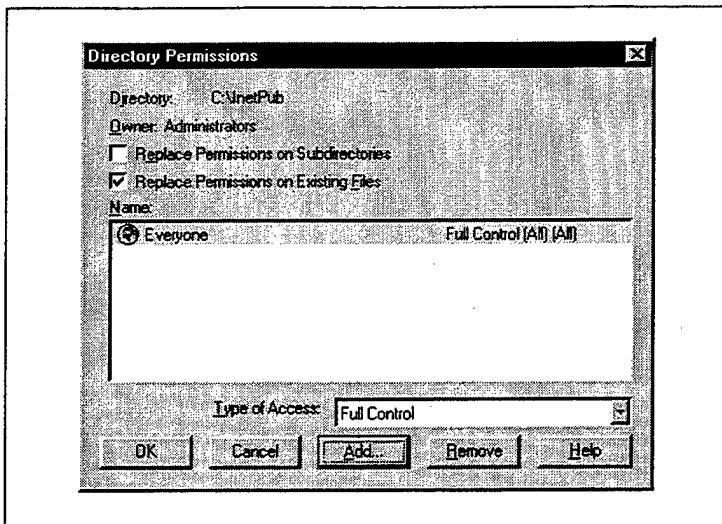
Figure 5.5. Using Windows NT Explorer to set directory permissions.

The permissions for the *InetPub* directory in this example are set
to *Everyone* with *Full Control*. If subdirectories have a more restrictive
set of controls, then Windows NT imposes the more restrictive set of
rules. For example, suppose that the HTML form is located in a
subdirectory called *Survey*. If access to this subdirectory is restricted
to *Survey Participants* and the *Everyone* access is removed, then it doesn't
matter that everyone has access to the *InetPub* directory. Only *Survey
Participants* have access to the *Survey* subdirectory, and Windows NT
verifies that the current client belongs to that group. Table 5.5
details the file system permissions available in NTFS volumes.

From Figure 5.5, it should be apparent that you can add users or
groups to the directory permissions and specify what type of access
they have. You should be aware that directory permissions are different
from share permissions, and when you create a share, by default
everyone is given full control permission.

PERMISSION	EFFECT
No Access	Prevents any access to the directory and its files even if the user has been granted share or file level full control.
List	Viewing, browsing directory, without access to files unless overridden by other file or directory permissions.
Read	Opening files and executing applications.
Add	Adding files and subdirectories without read access.
Change	*Add* and *read* permissions, plus delete.
Full Control	*Change* plus taking ownership and assigning permissions.

Table 5.5. Directory permissions [Ref. 26].

To identify the shared directories on *spitfire*, simply look in Windows NT Explorer for a blue hand icon underneath a directory name. Notice in Figure 5.6 that the *C, D, E, F,* and *G* drives as well as the *InetPub* and *Winnt* directories are shares. To view the share permissions, you can again use Windows NT Explorer, right click on the *IntetPub* directory, then select the *Share Permissions* tab. In general, you should note the following about permissions:

- Share permissions should not be changed. You can restrict access through file or directory level permissions that are more restrictive and will override the full control access that everyone has to the share.

- Notice in Figure 5.5 that you have the option to replace permissions on subdirectories. You should refrain from doing this because it will remove any previously imposed restrictions that you have in place. By replacing permissions on existing files, you are not changing the permissions of subdirectories or the files contained within them.

91
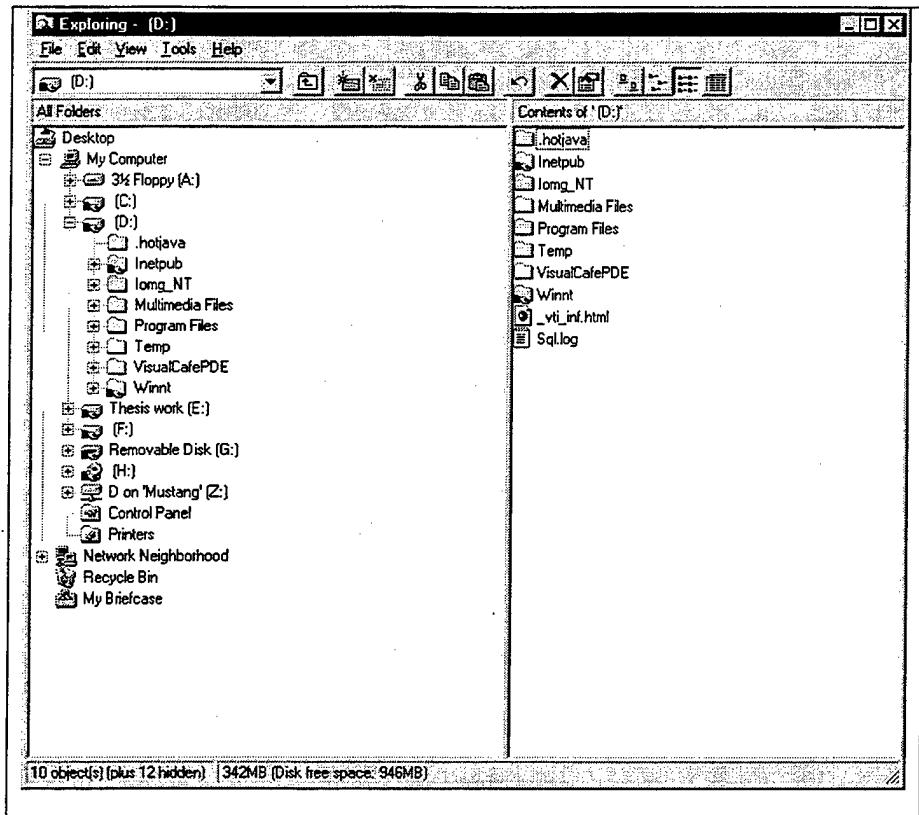
Figure 5.6. The *InetPub* share as viewed in Windows NT Explorer.

PERMISSION	EFFECT
No Access	Prevents access to the shared directory regardless of other allowed permissions.
Read	Allows viewing of contained files and directories, loading of files, and executing of software.
Change	All *Read* permissions plus creating, deleting, and changing contained directories and files.
Full Control	All *Change* permissions plus changing file system permissions and taking ownership.

Table 5.6. Share permissions [From Ref 1].

G. INTERNET INFORMATION SERVER (IIS)

IIS 4.0 is the latest version of Microsoft's web and File Transfer Protocol (FTP) server for the Windows NT Server 4.0 network operating system. It is an Internet standards-compliant Hypertext Transfer Protocol (HTTP) server that includes several other additional components which are used by the ACSA web site. In this section, we will explain the IIS installation process (to include where to get IIS and what components are included with the product), how to use the Microsoft Management Console and the HTML interface to administer the web site, how to examine server-generated log files, and how to implement security and authentication using IIS.

1.   Installation

IIS is included as part of the Windows NT 4.0 Option Pack and is available for download from the Microsoft web site at

[http://www.microsoft.com/windows/downloads/contents/updates/nt4optpk/default.asp]

Before you begin downloading files, you should ensure that your system meets the minimum requirements for IIS installation. The minimum system requirements, as noted by Microsoft, are as follows:

1. For Intel and compatible systems:

   - Pentium 66 MHz or higher; Pentium 90 MHz recommended

   - 32 MB of memory (RAM); 64 MB recommended

   - 200 MB of available hard-disk space

2. For Reduced Instruction Set (RISC) - based systems:

   - System with Alpha processor

   - 48 MB of memory (RAM); 64 MB recommended

   - 200 MB of available hard-disk space

3. CD-ROM drive

4. VGA, Super VGA, or video graphics adapter compatible with Windows NT Server 4.0

5. Microsoft Mouse or compatible pointing device

6. Service Pack 3.

Before you can download the Option Pack, you must register with Microsoft. The registration process establishes an account that you supply with a username and password. Microsoft sends a cookie to your computer to retain this information, so that any time you come back to their site, you can download other products without having to re-register. After registering, you will be redirected to the Windows NT Option Pack 4 download area. The Option Pack is almost 87MB in size, so you should ensure that you have a fast connection (such as a T1 line) before downloading the files. Beware! As you might expect, downloading the files over a 28.8 connection would take a great deal of time - almost 7 hours - and although a 56.6K connection is better, it still requires three-and-a-half hours of uninterrupted downloading.

The registration process will guide you to a list of several FTP sites you can choose from to obtain the files. Due to its size, the Option Pack cannot be downloaded in one bulk file. Rather, you must save 52 different files to disk and then run the program *install.exe* to begin the installation process. As you can imagine, this is a very tedious, timely process that can be avoided in three ways:

a) You purchase the Windows NT Server 4.0 Enterprise Edition, which includes the Option Pack.

b) You call Microsoft and pay them a tidy sum to receive the Option Pack on a CD.

c) You get a Microsoft Developer Network (MSDN) subscription, which includes all Microsoft operating systems, their BackOffice products, service packs, option packs, etc...

When you start installing the Option Pack, you will get a splash Screen as shown in Figure 5.7.
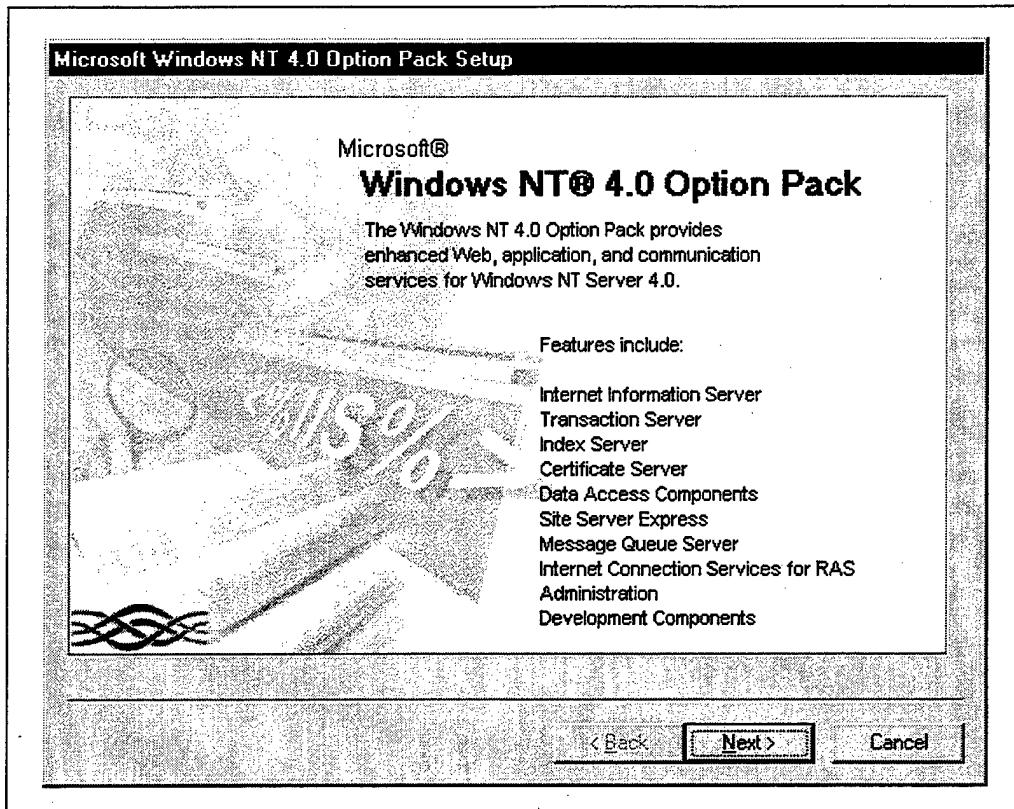
Figure 5.7. The Windows NT 4.0 Option Pack setup splash screen.

The installation wizard is self-explanatory and will guide you through several steps prompting you for information. If you accept the default settings, it will install the IIS WWW server and a FTP server to your computer. You can choose to add or remove additional components (see Figure 5.8) at a later time by going to *Start -> Programs -> Windows NT 4.0 Option Pack -> Windows NT 4.0 Option Pack Setup -> Add/Remove.*
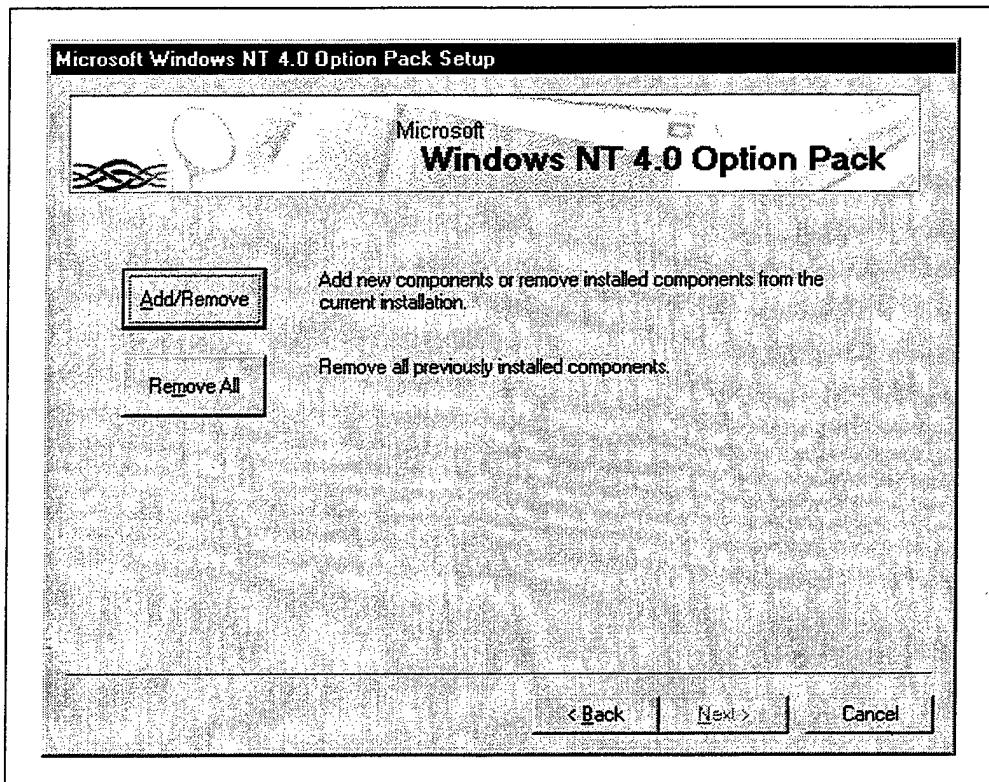
Figure 5.8. Running the Windows NT 4.0 Option Pack after it was already installed.

2.   NT Option Pack Components

IIS is distributed as part of the Windows NT 4.0 Option Pack. The Option Pack includes seven distinct software components, which are included in the list of features shown in Figure 5.7. We will briefly describe several of these components, and the most important ones that are related to web-site development are listed below for your reference:

- *Internet Information Server* (IIS) is Microsoft's robust, scalable web server for the Windows NT Server platform. With IIS, you can share documents and information across the Internet or an Intranet. Because it is integrated with Windows NT directory services, it offers a more secure means of publishing information and making it globally available.

- *Certificate Server* provides customizable services for issuing and managing certificates used in software security systems employing public-key cryptography. It performs a central role in the management of software security systems to enable secure communications across the Internet, corporate Intranets, and other nonsecure networks. You can use Certificate Server to generate certificates in standard X.509

format. This then allows you to authenticate servers and clients performing secure communications using the Secure Sockets Layer (SSL) protocol [Ref. 27]. The ACSA project uses certificates and the SSL protocol in combination with ASP technology to enable secure survey submissions.

- *Site Server Express 2.0* allows web-site administrators the ability to perform extensive web-site analysis. You can determine how a web site is constructed and analyze the contents of your web pages, or you can generate over twenty pre-defined reports that perform statistical analysis of the IIS log files. We will go into more detail on the use and features of this component later in this chapter.

- *Index Server* allows you to perform full-text searches and retrieve all types of information from your web site. Clients can search your site, and Index Server will return hyperlink matches that enable clients to jump to the information they are interested in viewing.

- *Transaction Server 2.0* creates a software environment in which any kind of transaction - an information transfer, a financial transaction, a database update, or any kind of activity that involves important or sensitive data - is guaranteed either to complete successfully or leave the data completely unchanged [Ref. 28]. Transaction Server is primarily used for distributed applications that adhere to Microsoft's Component Object Model (COM). Although the ACSA project is not a distributed application, future work may involve building it as such and taking advantage of this component to manage the application's database transactions.

Of course, no web site is complete without other services, and the NT Option Pack provides a rich array of supplementary services, to include FTP and the Network News Transport Protocol (NNTP).

The FTP service allows you to configure your computer as a file server, which means users (both those that have an account on your system and those who don't) can post and retrieve files. You should install and use FTP only if your users really need it, and even then, they should demonstrate a genuine need for the service. If FTP is installed, you should be extremely careful configuring it, and it is highly recommended that you disallow anonymous access. Additionally, remain cognizant of what files are being posted, as it could be the case that a malicious program has found its way into your system. We did install the FTP service on the *spitfire* computer for one reason:

- To afford us the capability to download documents from home.

97

As thesis documentation was completed or updated, it was placed in the *D:\Inetpub\ftproot* directory. From home, the developers were able to use a dial-in connection to gain access to the FTP service and download these files, or upload new ones. Because the Naval Post Graduate School has a firewall, outside access to ports that services use is only granted if the ports are explicitly listed as allowing access. The FTP service operates on port 21, and outside access to the FTP service is denied by the Naval Post Graduate School (NPS) firewall. To alleviate any concerns that the project sponsor may have, it is our intention at the conclusion of prototype development to stop the FTP service.

The NNTP service is similar to USENET newsgroups, but differs from them in the sense that IIS NNTP newsgroups are private online discussion forums. The ACSA project has three such newsgroups, as shown in Figure 5.9.
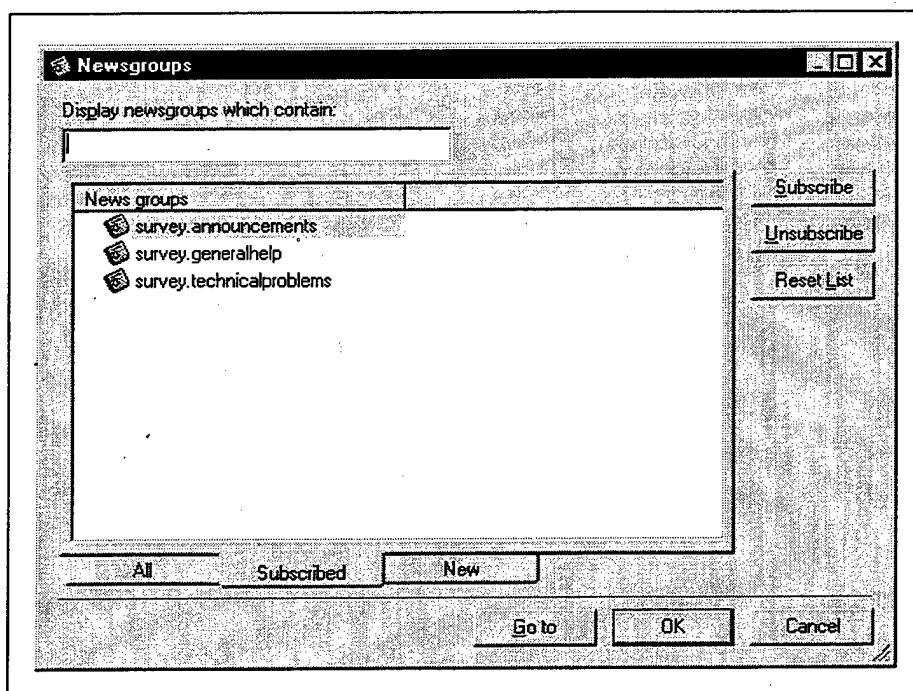
Figure 5.9. ACSA newsgroups.

You can post and read newsgroup messages. The purpose of newsgroups is to disseminate information to a wide audience that shares a common interest in the subject matter of that group. But newsgroups can also be used for administrative purposes. For example, suppose it was necessary to stop the IIS web server for maintenance. The system administrator could first post a message to the *survey.announcements*

group indicating that the *spitfire.avsafety.nps.navy.mil* web site would be unavailable, then shut down the web site but still provide newsgroup service to survey participants. The ability to run services independent of each other allows you various means of communicating with your clients. Newsgroups are located in subdirectories of the *D:\Inetpub\nntpfile\root* directory. More detail on how to configure the NNTP service and how to use your mail program to access the newsgroups will be explored later.

3. The IIS Web Service

The general idea of what a web server does and how it operates was explained at the beginning of this chapter. IIS offers two means of administering a web site:

a) You can use an HTML interface, known as HTMLA (or the HTML Administrator) to configure web server settings.

b) You can use the Microsoft Management Console (MMC), which provides what are known as "snap-ins" that can be used to configure your server and its services. Microsoft intends to make MMC the server configuration "nerve-center," and future releases of Windows NT (which will be part of the Windows 2000 product line), will incorporate the MMC concept.

To start the *HTML* interface, go to *Start* -> *Programs* -> *Windows NT 4.0 Option Pack* -> *Microsoft Internet Information Server* -> *Internet Service Manager (HTML)*. Your browser will open a page similar to the one shown in Figure 5.10.

Microsoft Internet Service Manager - Microsoft Internet Explorer

File   Edit   View   Go   Favorites   Help

Back   Forward   Stop   Refresh   Home   Search   Favorites   History   Channels   Fullscreen

Address http://localhost:3208/iisadmin/iis.asp

Links   Best of the Web   Channel Guide   Customize Links   Internet Explorer News   Internet Start

Internet Information Server   *Microsoft*

internet
service
manager

- localhost
  - Default FTP Site
  - Default Web Site
  - Administration Web Site

New Web...

Delete

Rename

Properties

Start

Stop

Pause

Resume

Master
Properties
Web...

Backup
Configuration

Local intranet zone

Figure 5.10. IIS HTML administration interface.

You should take note of the Uniform Resource Locator (URL)
displayed in the *Address* field of the browser window:
[http://localhost:3208/iisadmin/iis.asp]. *Localhost* indicates that you are opening the
page from the computer "local" to the location where IIS is installed.
The number 3028 is the port on which the administrator service runs.
Iisadmin is a virtual directory that is located separately from all your
publicly accessible web documents. How to create a virtual directory
using IIS beyond the scope of this chapter, suffice to say that you can
find the IIS administrator pages located in the subdirectory
D:\WINNT\system32\inetsrv\iisadmin. The contents of this directory are
listed in Table 5.7.

```
Volume in drive D has no label.
Volume Serial Number is D00E-BA02

 Directory of D:\WINNT\system32\inetsrv\iisadmin

[.]              [..]            blank.htm        calendar.asp
default.asp      default.htm     files            [htmldocs]
iiabout.asp      iiaccess.asp    iiacct.asp       iiacsshd.asp
iiacssls.asp     iiaction.asp    iiadm.asp        iiadmbd.asp
iiadmhd.asp      iiadmls.asp     iiamap.asp       iiamaphd.asp
iiamapls.asp     iiapp.asp       iiapphd.asp      iiappmn.asp
iiauth.asp       iibind.inc      iibkup.asp       iibkuphd.asp
iibkupls.asp     iibkupset.asp   iibkuptl.asp     iibody.asp
iicache.asp      iichkpath.asp   iicomm.asp       iicomp.asp
iidef.asp        iidoc.asp       iierr.asp        iierrhd.asp
iierrls.asp      iifilt.asp      iifilthd.asp     iifiltls.asp
iifixpth.inc     iifmsg.asp      iifnav.asp       iiftp.asp
iifvd.asp        iifvddir.asp    iifvdhd.asp      iifvdnt.asp
iifvdrd.asp      iifvs.asp       iigtdata.asp     iihd.asp
iihdn.asp        iihdr.asp       iihdrhd.asp      iihdrls.asp
iilist.asp       iilog.asp       iiloghd.asp      iilogmn.asp
iimime.asp       iimimehd.asp    iimimels.asp     iimlti.asp
iimltihd.asp     iimltils.asp    iimnu.asp        iinav.asp
iipath.asp       iipathhd.asp    iipathls.asp     iiperf.asp
iipop.asp        iipopcl.asp     iipopcl.htm      iipophd.asp
iipoptl.asp      iiput.asp       iiputls.asp      iirename.asp
iirte.asp        iirtehd.asp     iirtels.asp      iis.asp
iiscript.asp     iisec.asp       iisechd.asp      iisecls.asp
iisess.asp       iiset.asp       iiset.inc        iislider.asp
iisnew.asp       iisrvls.asp     iistat.asp       iito.inc
iitool.asp       iivd.asp        iivddir.asp      iivdhd.asp
iivdnt.asp       iivdrd.asp      iivs.asp         iiwww.asp
[images]         [isadmin]
[jsbrowser]
             111 File(s)        452,639 bytes
                          973,870,080 bytes free
```

Table 5.7. IIS HTML administration files located in the virtual
directory *IISAdmin*.

If you explicitly try to administer the web site from a remote
location using the URL [http://spitfire.avsafety.nps.navy.mil:3208/iisadmin/iis.asp], you
will receive an error message indicating that access to the *HTMLA* is
restricted to the *localhost*. The 403.6 forbidden message (see Figure
5.10) provides detailed instructions on how the system administrator
can enable remote access to the *HTMLA*. The only time this feature
should be enabled is when the system administrator is going to be
offsite for an extended period of time and has a need to remotely
administer a web site served by IIS.
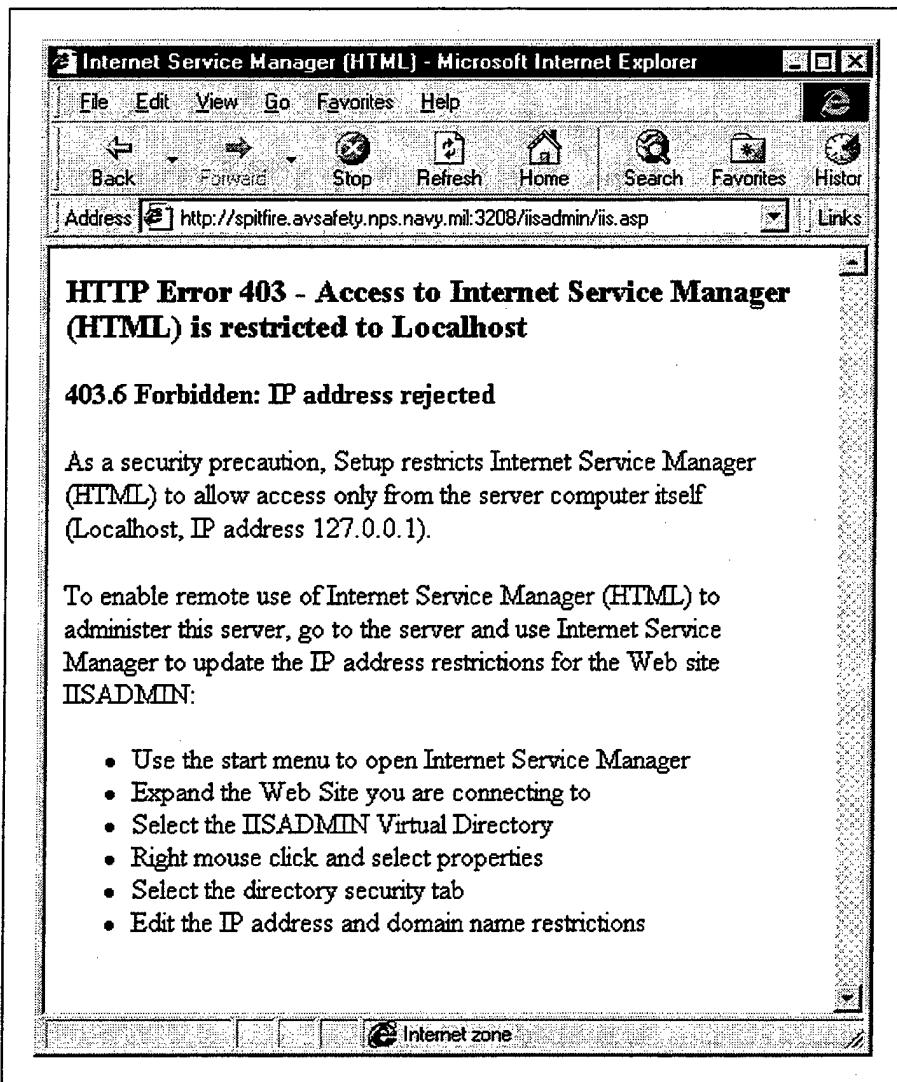
Figure 5.11. Trying to access HTMLA from a remote computer.

If you expand the *Default Web Site* shown in Figure 5.10, IIS will proceed to load your web site's virtual directory information, and the directory structure of everything located below *D:\Inetpub\wwwroot*. This takes several seconds, and once IIS is ready, you should end up with a view of your web site similar to Figure 5.11. As you peruse the contents of the site, take some time to look at the left side of the Internet Service Manager. Here, you will find the commands *New, Delete, Rename,* and *Properties,* which allow you to create or delete web sites (an IIS web server can host multiple web sites), rename them, or look at their properties. Because the ACSA project is the only web site hosted on the *spitfire* server, we will avoid discussing how to use these IIS features. Do notice, however, that from here the administrator can *Start,*

Stop, Pause, and *Resume* the web service, examine web service properties and change them by using the *Master Properties* link, or backup an entire web site.

It is fairly obvious that IIS provides an extensive set of features that can be used to create very complex sites. Fortunately, administering an IIS web site with the *HTMLA* is straightforward, and Microsoft provides extensive browser-viewable documentation (for IIS and the other Option Pack components) which you can access at [http://localhost/iishelp/iis/misc/] . When you attempt to access this documentation, IIS will query you for a username and password, which should correspond to one of the accounts listed in the *User Manager for Domains*.
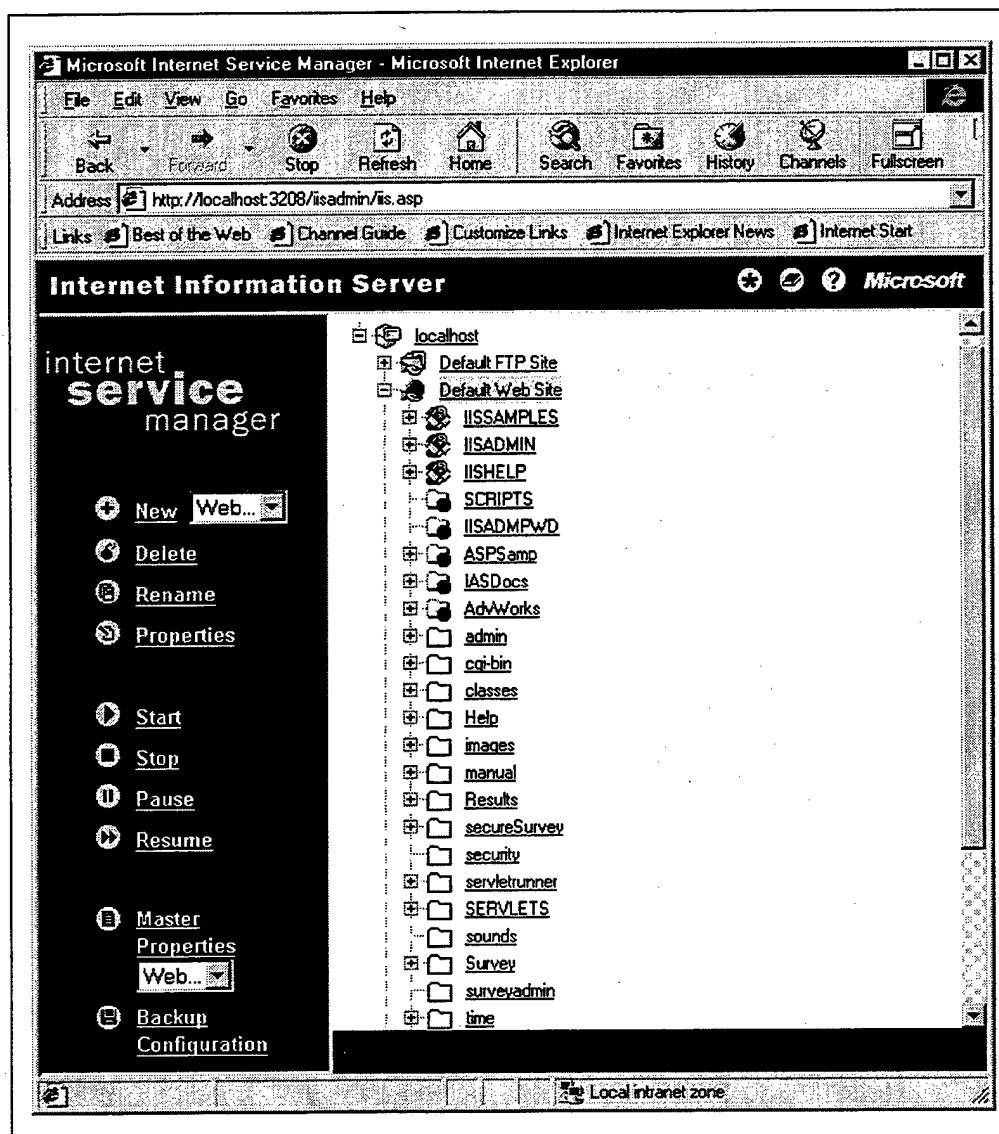
Figure 5.12. An expanded view of the default web site's directories.

As with anything else, experimenting with the *HTMLA* will not only gain you familiarity with the tool, but also make you an expert on how to run your web site and what you can do with IIS. From here, let's click on the *Master Properties* link. Notice in Figure 5.11 that there is a drop down selection box below this link. This box indicates what service we are going to examine, and the default selection is set to the IIS web service. You will now be presented with a view similar to that shown in Figure 5.13.
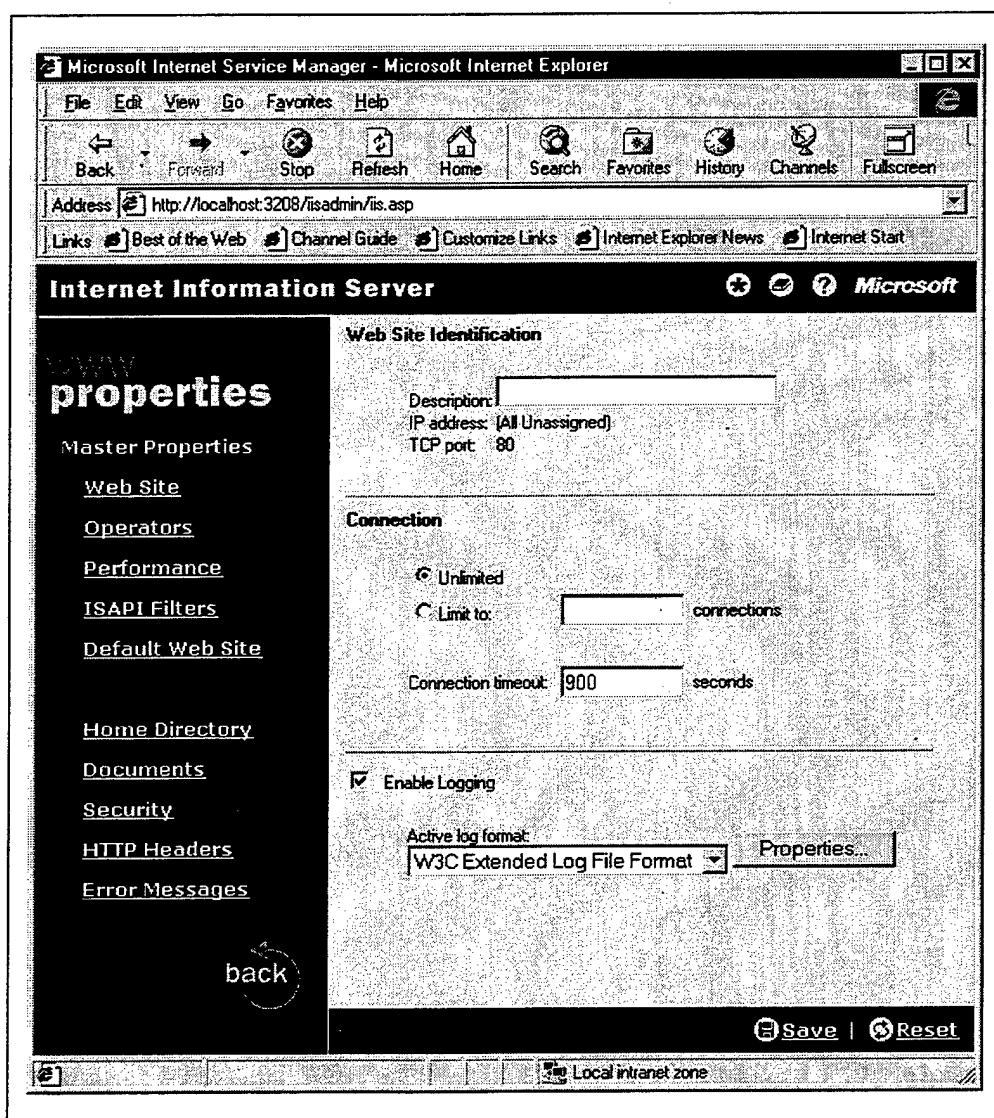
Figure 5.13. Web site master properties.

You now have even more options than before, and some explanation

as to what you can do from here is necessary. First, notice that you
are currently allowing an unlimited number of client connections. If
you find that your site is extremely busy and performance is degraded
due to a large number of simultaneous client requests, you can limit
the number of connections (the ACSA prototype system should never
experience this problem due to its limited audience size). Another
connection property setting you can set is *timeout*, which indicates how
long IIS will wait before disconnecting an inactive user.

And if you want to maintain information about your visitors, you
can enable logging. There are several file formats you can choose from.
The default log format is set to *W3C Extended Log File Format*. This ASCII-
formatted file can be viewed in a text editor such as *Wordpad*, or you
can import it into *Site Server Express* and generate reports based on an
analysis of these cryptic log files. You can selectively choose what
information to record by clicking the *Properties* button, which allows you
to set both *General* and *Extended* log file properties. Figure 5.14 shows
what features can be customized.
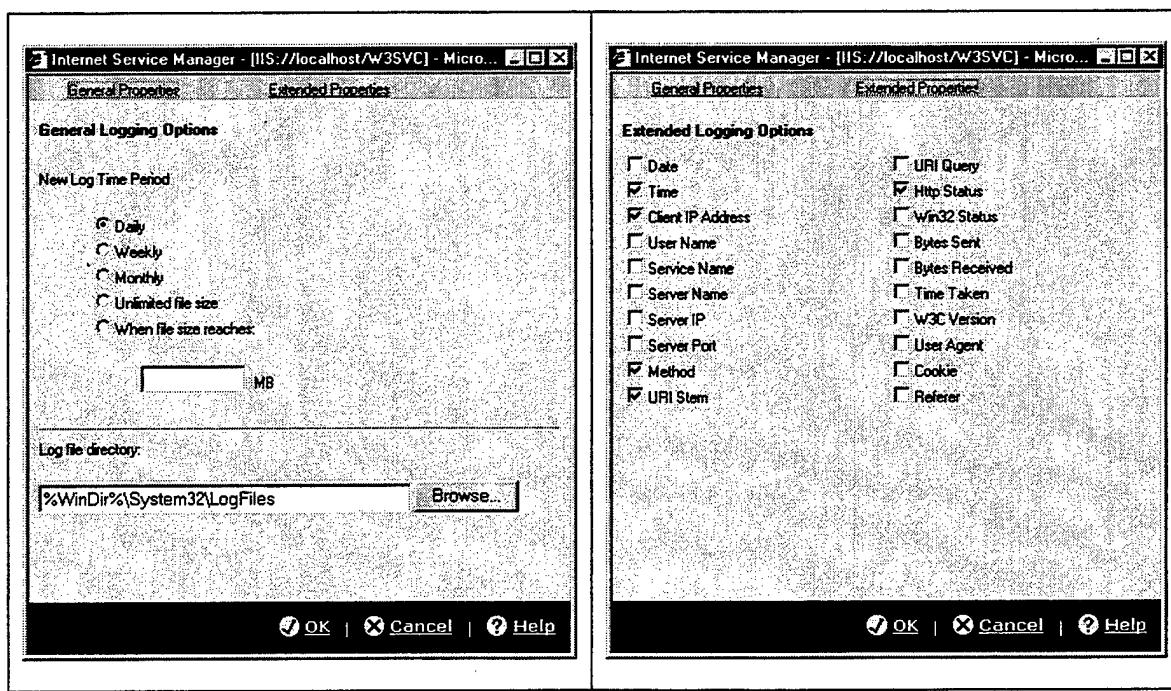
Figure 5.14. Customizing the log file.

All log files can be found in the *D:\WINNT\system32\logfiles*
subdirectory. The ACSA program administrator should examine these log
files on a daily basis (they are saved in the format exYYMMDD.log,

where YY indicates the year, MM the month, and DD the day). Your main concern is to look for unusual activity and to note the Internet Protocol (IP) address the activity originated from. Once you have this address, you can use a utility such as *nslookup* to resolve the domain name of the IP address. For instance, suppose I saw some unusual activity coming from 206.39.12.4. I would proceed to open a MSDOS prompt and at the command prompt, type *nslookup 206.39.12.4*. The results from this query are illustrated in Table 5.8.

Server: nps.navy.mil
Address: 131.120.254.52

Name: www.dfas.mil
Address: 206.39.12.4

Table 5.8. Result of executing *nslookup 206.39.12.4*.

If the user's activity is suspect, further correspondence between system administrators at NPS and the activity in question (e.g. DFAS in this case) should resolve the issue.

Other WWW properties you can manage include:

- *Operators,* which allows you to specify what additional NT user accounts have web-site administrator privileges.

- *Performance,* which you can use to tune the web server based on the number of hits expected per day.

- *Home Directory,* which is used to manage virtual directory properties.

- *Documents,* which allows you to list a default document name that will be loaded by the web server every time a directory is accessed but no file name is provided.

- *Security,* which allows you to implement directory and file security permissions. More on this shortly.

- *Error Messages,* which allow you to choose what file will be returned in the event that an error status code needs to be returned.

4.   Web Site Directory Security

One of the early distinctions made in the ACSA project was that different users needed different degrees of access to the web site. It was common belief that everyone should be allowed to submit surveys (within reason), but only Commanding Officers should be allowed to view the aggregate database results. In order to separate these two classes

106

of users, we created user accounts that were authorized different permissions, i.e. they had different rights. Then, as we created the web site, we divided the content of directories into logical areas that were later assigned permissions as described earlier in the section on Windows NT Security in this chapter.

The manner in which you use the *HTMLA* interface to assign permissions is discussed here. You should have the *HTMLA* open, and highlight the *Results* subdirectory. Click *Properties* on the left side of the *Internet Service Manager*, and you will be guided to another page that allows you to select *Security*. After the *Security* button, a third page will load that presents you with three options:

a) **Anonymous Access and Authentication Control**
Click *Edit* to configure your Web server's authentication and anonymous access features. Use these features to instruct your Web server to confirm the identity of users before granting access to restricted content. Before your server can authenticate users, you must first create valid Windows NT user accounts and then configure Windows NT File System (NTFS) directory and file permissions for those accounts.

b) **Secure Communication** allows you to configure your web server's Secure Sockets Layer (SSL) 3.0 secure communications feature. The ACSA project provides a secure ASP application that takes advantage of SSL via a key (or digital certificate) exchange between the server and the client's browser.

c) **IP Address and Domain Name Restrictions** can be used to increase the granularity with which external organizations have access to the IIS web site. We could, using this option, enumerate the IP addresses of computers that can have access to the web site, but this procedure would be time extremely time consuming and troublesome.

Because the *Results* directory does not require options b or c, you should go ahead and click on Option a. Figure 5.15 illustrates what you should see. You will notice that we explicitly disallow anonymous access to this directory and the checkbox next to *Basic Authentication* is marked. What this means is that if a client requests any file located in the *Results* directory, he will have to provide a valid username and password that corresponds to a user account on the *spitfire* machine. Without the correct information, access to the *Results* applet will be denied. Unfortunately, this means that anyone who has a user account

will be able to access the information contained in this directory, so to restrict which users have access from those that don't, we have to use either Windows NT Explorer or the Microsoft Management Console (MMC). If you run Windows NT Explorer, right click once on *D:\Inetpub\wwwroot\results*, select *Properties*, click on the *Security* tab and select *Permissions* (recall that this procedure was described earlier in the chapter), you will find a list of users that have access. Specifically, all *Administrators*, the user accounts *Commander* and *DocC*, and the *SYSTEM* process all have access (**WARNING**: You should never delete or remove *SYSTEM* from the list of users that have access, as this special built-in account refers to the operating system and not a specific user).
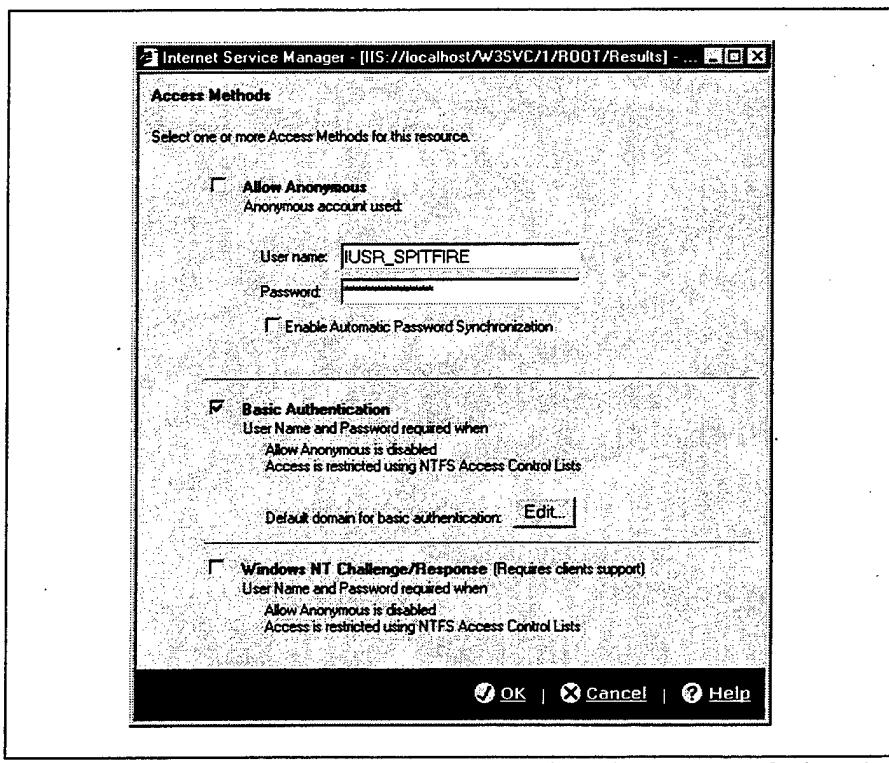
Figure 5.15. Anonymous Access and Authentication Control in the HTMLA.

Other directories that have permissions set are the *Survey* directory (access is granted to the *Administators* and *Survey Participants* group, the user account *DocC*, and *SYSTEM*). Finally, the *wwwroot* directory, which contains all of these subdirectories, is itself protected with the permissions shown in Figure 5.16.
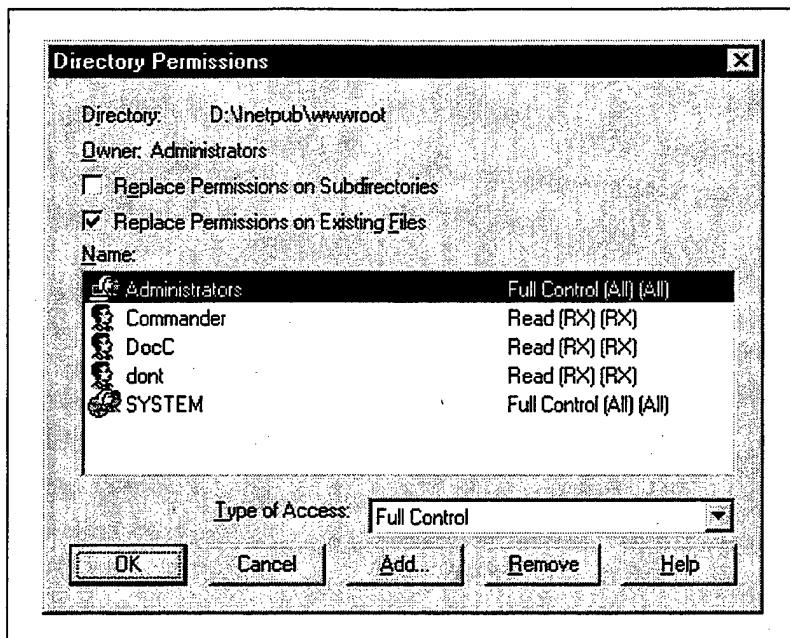
Figure 5.16. Permissions on the *D:\Inetpub\wwwroot* directory.

The implications of this restriction mean that the contents of the web site are not open to the general public. This approach was taken in large part due to a new set of regulations promulgated by the Secretary of the Navy regarding worldwide web use. A summary of these requirements can be found in Appendix J. To preclude having to overhaul the web site and meet each and every requirement, it sufficed to put a security disclaimer on the main page and secure it from public access. You should also note that this web site has not been registered, nor should be registered, with any general search engines (such as Lycos, Excite, Hotbot, or Webcrawler). By not registering this site, you greatly reduce the risk of intruders attempting to penetrate your system.

5.   Using Secure Sockets Layer (SSL) for Securing Web Site Communications

Up until this point, we've addressed security in a very limited way. Recall that we talked about the need to prevent one client from entering multiple surveys, and we partially addressed this issue by creating and using the *Num_List_DB* database (a client could potentially enter all surveys for the block of access numbers assigned to his squadron). This utility database adds integrity to the survey process, but it offers nothing in terms of privacy and doesn't address system availability. It is the former area, privacy, that we examine here.

109

You might be wondering how all of the information is sent from the client to the *spitfire* machine. When a client submits his survey, it is bundled into TCP/IP packets and sent over the Internet. The information in these packets is unencrypted clear text, and can be analyzed by almost anyone with a packet sniffer (a program that can analyze TCP/IP packets travelling across a network). This information is probably uninteresting and unintelligible without the actual survey, but nonetheless, clients are sometimes ill at ease with sending information in an insecure manner. For this reason, the ACSA project takes advantage of digital certificates and the Secure Sockets Layer (SSL) protocol to encrypt information.

The SSL portion of the ACSA web site uses a form of cryptography known as public key encryption. Using the *Certificate Server* component of IIS, we can generate a public and private key for the server's web service. These 40-bit keys do not provide the strongest encryption means available (128-bit keys provide more protection), but suffice for the purpose of this project. Now, if a client wants to access the secure portion of the web site, he simply types in a URL that begins with *https* rather than the traditional *http* prefix, e.g., [https://spitfire.avsafety.nps.navy.mil].

When the client does this, his browser attempts to communicate with the server on port number 443. (Standard *http* requests default to port 80 of the target machine). If the server is supporting the SSL protocol on this port, it will send its public key back to the client. The client's browser indicates that it is receiving this key by showing a lock icon that is colored yellow. The browser uses this key to encrypt the information it is going to send to the server. The only way the information can successfully be decrypted (other than through brute force techniques) is for the server to apply its private key on the encrypted data. For this reason, administrators should ensure that the private key is carefully protected. If it is ever compromised, administrators should attempt to find out why, correct the problem if possible, and immediately generate a new certificate.

You can generate keys for the various services (WWW, FTP, NNTP, etc...), by first starting the Microsoft Management Console (MMC) (see Figure 5.17). MMC is found by selecting *Start->Programs->Windows NT 4.0 Option Pack->Internet Information Server->Internet Service Manager.* Once the

program is running, you can select the *Key Manager* icon in the programs menu and should see a new application window similar to the one shown in Figure 5.18.
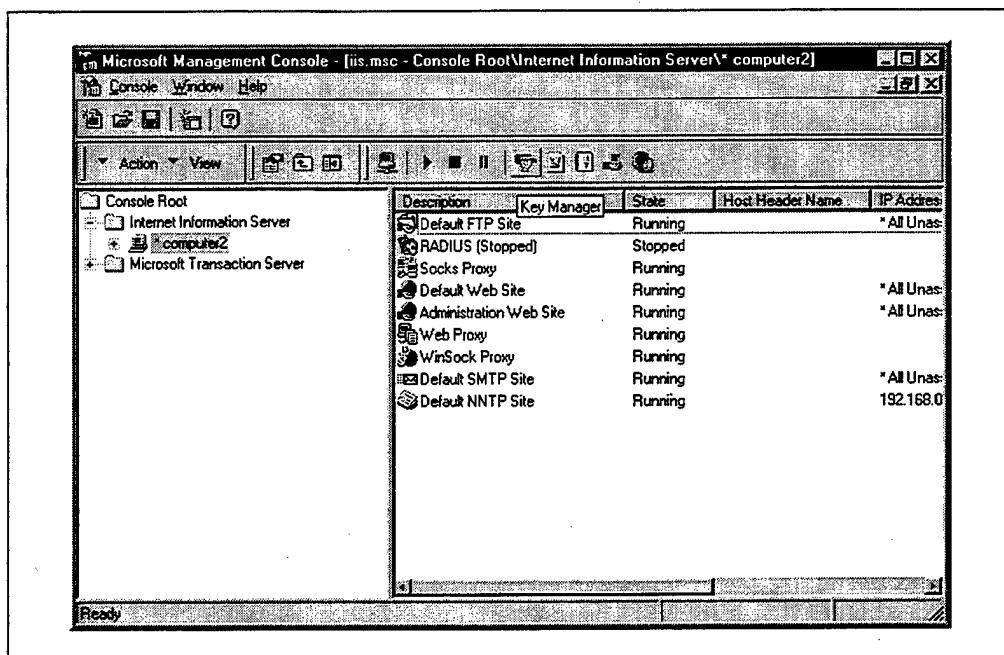
Figure 5.17. *Key Manager* icon in MMC.
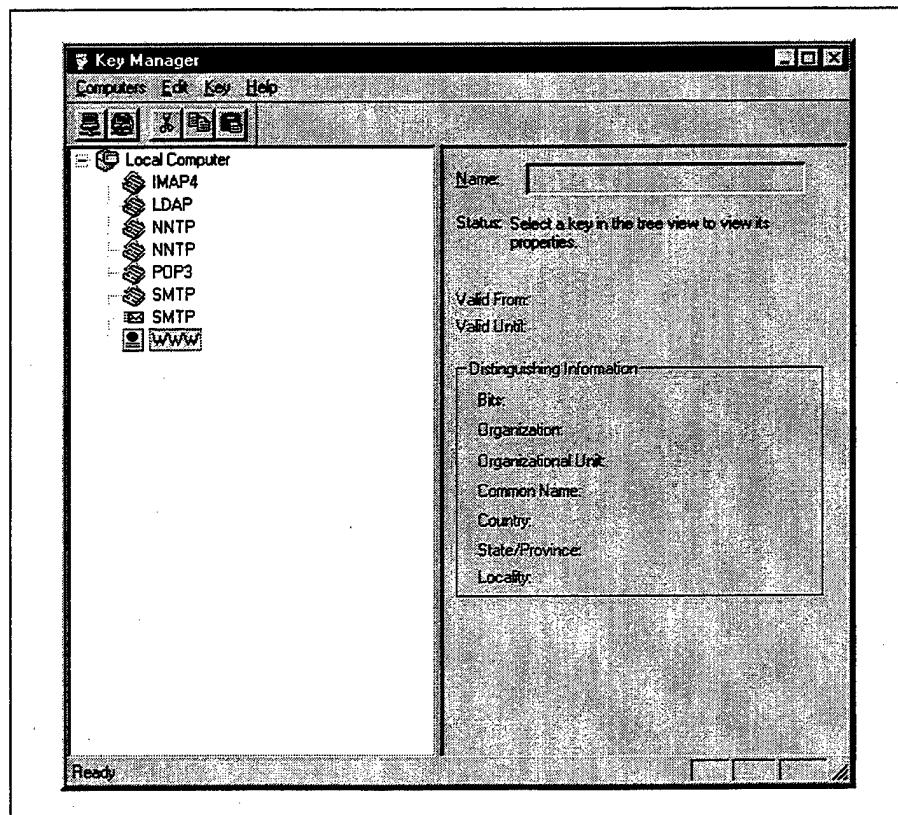
Figure 5.18. The *Key Manager* application.

You can now highlight the service that needs a key, and generate one by selecting *Key->Create New Key* from the menu, or by right-clicking on the service and selecting *Create New Key*. To guide you through the process of generating a key, we will create one for the WWW service and outline each step that is required to successfully create a key. You should first ensure that the WWW service is highlighted, as shown in Figure 5.18, and then select the option to create a new key by either of the means previously described. This should bring up a new dialog box with the title "*Create New Key*" (see Figure 5.19). Here, you have the option to either save the certificate to a file (and later forward it to a Certificate Authority (CA) such as Verisign), or to send the request to an online authority. We are going to choose the latter option and send the newly generated certificate to Microsoft's *Certificate Server* application. This application will generate a key that expires one year from the day it was created, which effectively means that we are acting as our own CA. Because we are the CA and have not had our identity confirmed by a third party, clients will receive our certificate but not know if it is indeed us that generated the key that the server is sending. Though this is relatively insignificant for the ACSA project, it would definitely deter clients from doing business with a company that was processing credit-card transactions for their products. Should you desire an identity-validated certificate, you should send your certificate request to an online CA for verification, and then use the *Key Manager* application to bind the new certificate to whatever services you intend to use it with.
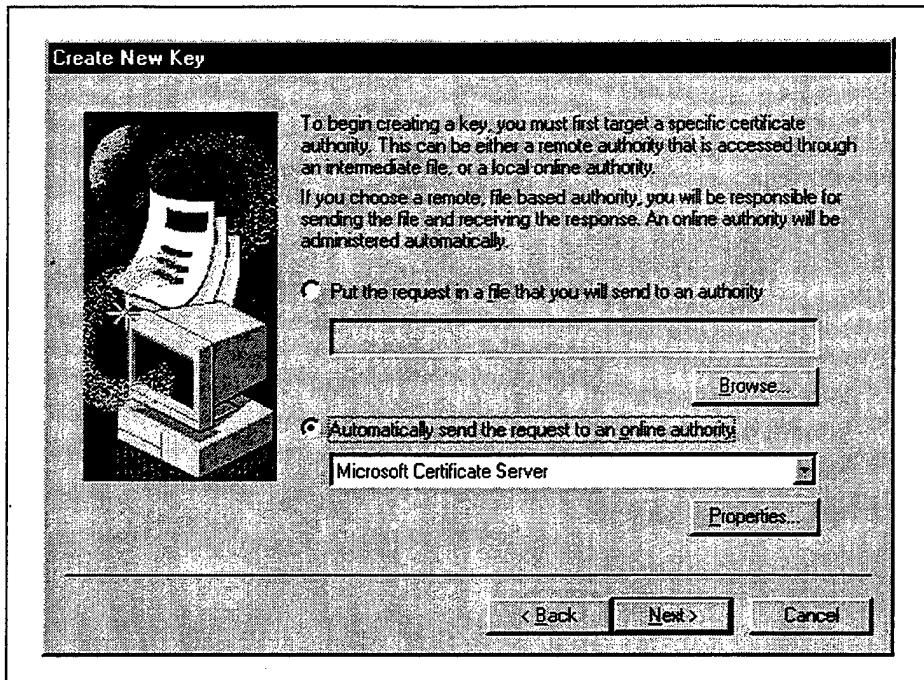
Figure 5.19. Creating a new key and binding it to the WWW service.

The next several screens allow you to specify the properties of the key, such as its name (or common name), bit length (the default is 512), and the organization it belongs to and the organization's location. When you are finished, the key is forwarded to the *Certificate Server*, which will process it and return you back to the *Key Manager* application. It is here that you should now see a key icon below the WWW service. The key has a name, and on the right, you should notice the key's status (if it is usable or not), information on when it was created and when it will expire, and distinguishing information about the key. Figure 5.20 illustrates what you should see when your new key has been successfully created and bound to the WWW service.
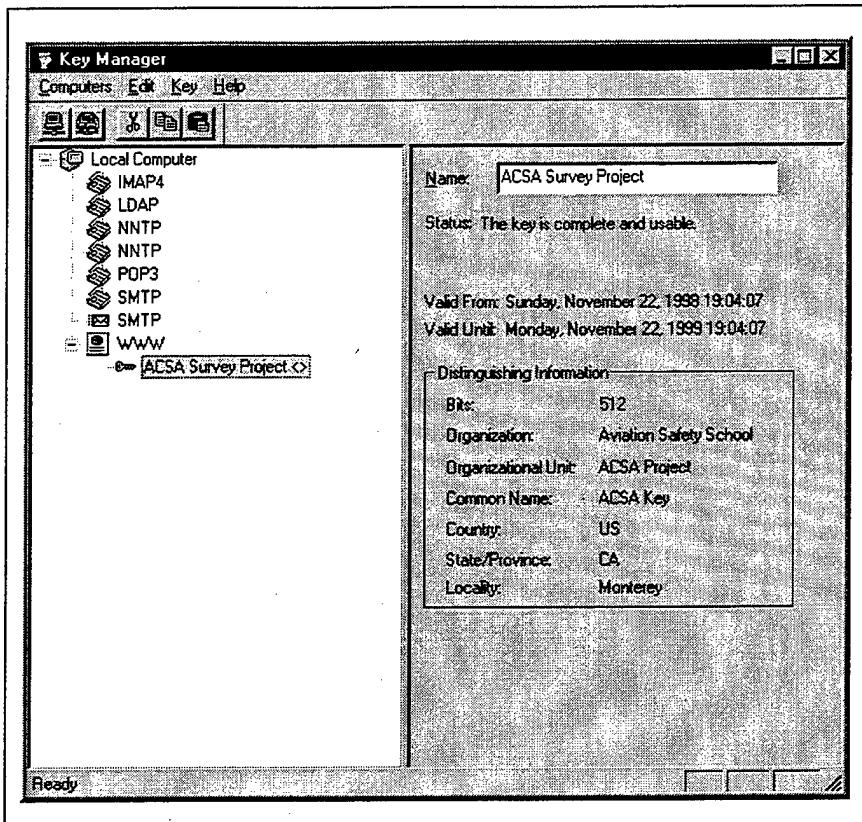
Figure 5.20. Successfully binding a key to the WWW service, as seen in the *Key Manager* application.

With the new key, you can now access the web site in a secure manner - simply use the *https* prefix followed by the server's domain name (*spitfire.avsafety.nps.navy.mil*). But suppose there is the more stringent requirement that certain portions of the web site can **ONLY** be accessed in a secure manner. Such is the case with the ACSA project, where we want the administrator's pages and the ASP version of the HTML survey to be accessed in this manner. To do this, we simply put these parts of the web site in their own subdirectories, and then use the HTMLA to configure each subdirectory for secure communications access.

6.   **Network News Transport Protocol (NNTP)**

We briefly mentioned newsgroups earlier in this chapter and the three that were related to the ACSA project. Installing the NNTP service is easy to do. At any time after you have installed the Windows NT 4.0 Option Pack, you can rerun the setup program and add components or remove previously installed ones. Since our initial installation did not include the NNTP service, we had to add it. Once this was done, the

114

NNTP service was configured by using an HTML interface very similar to the WWW HTMLA previously described. You can find the NNTP HTML interface by going to *Start->Progams->Windows NT 4.0 Option Pack->Internet Information Server->Microsoft NNTP Service->NNTP Service Manager (HTML).* Figure 5.20 shows the NNTP HTML administrative interface. From here, you can manage the NNTP service in the following ways:

- specify the port that the service runs on (the default is 119), the number of simultaneous connections that can be made, the maximum article size that a client can post

- configure the NNTP service to use a Simple Mail Transport Protocol (SMTP) server to forward articles to newsgroup moderators

- set newsgroup expiration policies, e.g. how long an article should remain posted before it is deleted from the newsgroup

- determine what clients are using the NNTP service by examining session information

- configure NNTP security settings

You should take some time to use this interface and become familiar with NNTP service settings. Once the newsgroups have been created and are ready to use, the final step in the process is to use a newsgroup reader or configure your email client (such as Microsoft Outlook Express) to use the NNTP service (the name of the NNTP server is *spitfire.avsafety.nps.navy.mil*). For more information on how this is done, please consult the documentation that came with your software.

115
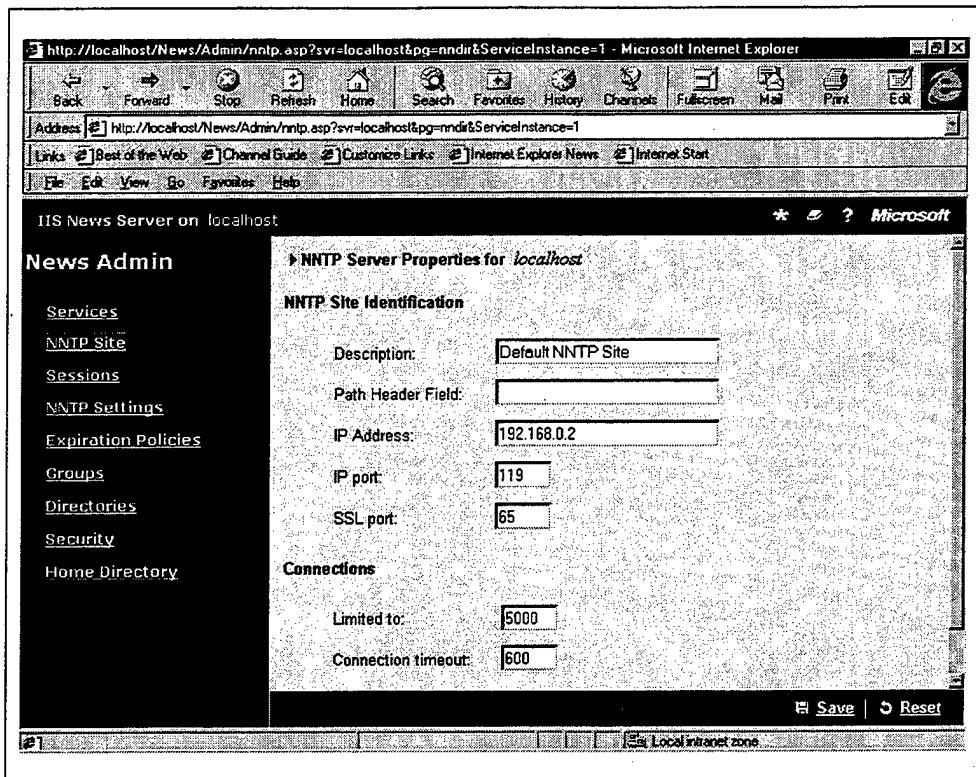
Figure 5.21. NNTP Server Properties.

7. Configuring Services Using the Microsoft Management (MMC) Console

The Microsoft Management Console (MMC) is a standardized administrative interface that is more than just a utility for configuring IIS. In fact, future versions of the Windows NT operating system (now known as Windows 2000), will use MMC as the central means for controlling, managing, and administering all NT services. MMC was prereleased as part of the Windows NT 4.0 Option Pack to allow users to acclimate themselves to the new environment well before the next NT OS is released.

The ACSA project runs on Windows NT 4.0 Server software and has the Option Pack already installed. The HTML interface used to configure various services has already been discussed, but we never considered what happens if the IIS web server software is not running or becomes corrupted. It should be fairly obvious that in this case, the HTML interface to configure your Internet-related services is no longer available. At first thought, we may be tempted to reinstall the Option Pack and hope for the best, but the better solution to troubleshooting and making configuration changes is provided by MMC.

116

You launch MMC by going to *Start->Programs->Windows NT 4.0 Option Pack->Microsoft Internet Information Server->Internet Service Manager.* MMC will run, and you should see something similar to that shown in Figure 5.22.
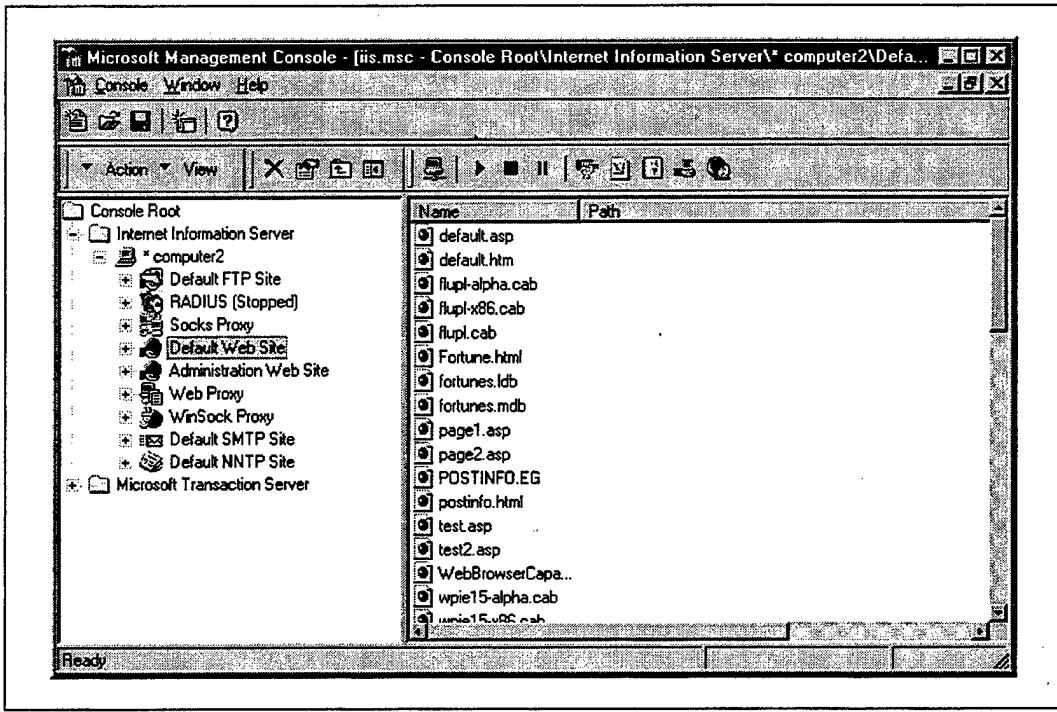
Figure 5.22. Microsoft Management Console (MMC) used to configure Windows NT services.

From Figure 5.22, it is apparent that there are several services for *computer2* that are installed and running: e.g. FTP, HTTP, SMTP, NNTP, as well as some proxy services. Notice that the RADIUS service is stopped. You can start, stop, and pause services by highlighting them, clicking on the right mouse button, and selecting the appropriate action. Alternatively, the black triangle, square, and two parallel lines can be clicked to start, stop, or pause a service respectively.

By far the greatest number of configuration changes that can be made are to the web service. Figure 5.23 shows the property sheet that appears when you highlight the web service, right click, and select *Properties* from the pop-up menu that appears. Among the many changes that you can make are:

- You can change the web service to a non-standard port number, thereby restricting access to those who know the new port number. The SSL port can likewise be changed.

117

- You can grant access based on Windows NT user accounts.

- You can customize HTTP headers (e.g. the 404 File Not Found message) and have IIS return the page to the client in the event of an error.

- You can modify directory security permissions.

* Changes are committed only after you exit the program. MMC will prompt you "Do you want to save your settinngs to iis.msc?", to which you should click *OK*.
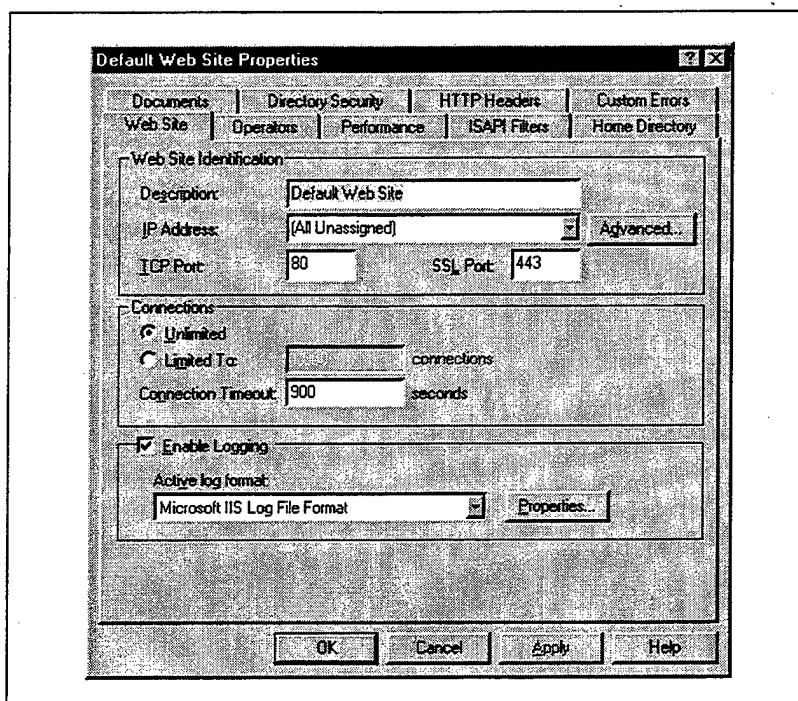
Figure 5.23. Default web site properties as configured through MMC.

The other services also have various properties that can be configured through MMC. It is worthwhile to take some time and examine each and every one of these properties. If you are unsure of what a property does, consult the Windows NT Option Pack 4.0 documentation at [http://localhost/iishelp/iis/misc/default.asp] before proceeding.

8.   **Ports Used by the ACSA Project**

PORT NUMBER	SERVICE
80	WWW
119	NNTP
443	SSL
1443	SQL SERVER
8080	SRVLETRUNNER

Table 5.9. ACSA ports and services.

118

H.   CONCLUDING REMARKS

The Windows NT 4.0 Option Pack comes bundled with a set of powerful, information-management components that provide significant support to Enterprise Information Systems (EIS). The ACSA project uses many of these components to provide a robust, scalable site that can deliver information to clients in many different ways. The configuration of these services is an extremely important topic for future developers and administrators. Proper lifecycle support can only be provided if the individual(s) responsible for this task are intimately familiar with the concepts presented in this chapter. The ACSA web site has been successfully field-tested and is ready for Navy-wide use, and although it can pretty much run itself in a standalone manner, it still requires human intervention to comb through IIS log files, make changes to HTML pages, or change any of the underlying Java code for product enhancement purposes.

VI. CONCLUSION

A. CLOSING REMARKS

The purpose of this thesis was to design, develop and test an automated, cost-effective prototype model for the Aviation Command Safety Assessment (ACSA). Our approach to this task was to integrate existing Commercial Off-the-Shelf (COTS) technologies with our technical expertise. In the end we developed a system for a cost of less than $5000. We think we met this initial challenge, but the question remains, is how easy will it be for future developers to revise or improve upon this project? To facilitate these efforts, we have attempted to lead the reader through our design and development thought processes, with an added emphasis on documentation. From this perspective, then, another measure of our success will be the ease with which future developers can rely upon our documentation as presented in the previous five chapters.

At this point, we should note that there are several minor revisions which are not accurately represented in this document. Although we were continually making minor code enhancements based upon usability and testing feedback, we did not go back and change our write-up for minor revisions. Consequently, the following ACSA web site documentation differs from the actual ACSA web site in the following areas:

- **Expanded survey participants** to include all squadron aircrew as participants, not just officer aircrew. The granularity with which all aircrew positions are represented is fairly large, but expanding the scope of the project to consider all aircrew was a last minute modification.

- **Elimination of any actual squadron number reference** to further protect the anonymity of each squadron. With this revision the database will not contain individual squadrons, rather communities with access numbers assigned by the ACSA program administrator.

The true measure of success for the automated ACSA web site will be the desirability and usability of this new process from both the client and administrator perspective. The site must be accessible, easy to use, deliver valuable information in a timely fashion and not create an administrative burden for the program administrator. On 9 November 1998, we presented our thesis to the Human Factors QMB via video-teleconference. Our audience included senior officers and civilian

representatives from all echelons of naval aviation and was viewed from seven different locations throughout the world. The brief included a short thesis overview followed by an online real-time site tour. Then, on 18 and 19 November 1998, we went into the field for squadron usability testing and evaluation. During this visit, we briefed and configured the computers at three volunteer squadrons from AIRPAC. In every case the prototype system was enthusiastically received. In fact, the participating squadrons wanted to begin submitting data immediately. Unfortunately, the Internet connection from these squadrons was extremely slow, rendering the applets essentially unusable. Although we cannot control these client-side variables, we believe these limitations will be corrected, thereby enabling participating clients to take full advantage of what the web site has to offer.

B. AREAS FOR FUTURE RESEARCH

As in any research development effort, we were constantly experimenting and learning about the technologies we were working with. Reflecting back upon several of our design decisions, we recognize there might have been a better way to implement some tasks. Often, we revised our design and/or code, but on many occasions, we did not go back to affect this new perspective. At this point, we feel any program revision should include:

- **Making components for the applets:** The code in this project has been written specifically for this ACSA project model as it exists today. Seemingly minor questionnaire changes, e.g., changing the number of questions or the respondent's options, will require considerable recoding and programmer effort. Putting the questions into a readable, ASCII-text file which could then be used by the applet help create an independent separation that would allow questions to be changed without program code having to be rewritten. The use of JavaBean components would further aid in this separation and would result in the creation of independent modules with well-defined tasks rather than having one OOP program do everything.

There are a number of additional areas that could further enhance the ACSA web site. Areas for future research include:

1. **E-mail processor:** The current model requires an Internet connection for submitting questionnaire data. Expanding the model to receive e-mail submissions would enable squadron

detachments afloat or those without direct Internet access the opportunity to submit questionnaires.

2. **On-line individual unit responses**: Probably the most sought after program enhancement is on-line squadron feedback for the individual unit commanding officer. In our presentations to each squadron commander, they specifically requested this feature. While individual online results are desirable from the client perspective, their implementation from the ACSA administrator's perspective is considerably more complicated.

3. **Automated alerts and data monitoring**: Although the specific implementation options are still being discussed, the current web site requires the ACSA administrator to physically analyze the database. Incorporating an automated rule-based data monitoring process would alert the ACSA program administrator of any alarming trends in real-time.

4. **Operational Risk Management (ORM) controls**: The current model only performs the first step in the five-step ORM process: hazard identification. Expanding the model to include hazard assessment and then recommending risk management controls is the next logical extension to this model.

5. **Active platform "push technology"**: Related to the discussion in 4 above, when the decision is made to initiate a survey the ACSA administrator could "push" this notification to every squadron along with their individual survey submission access number and individual survey submission numbers.

C. ACSA PROJECT IMPLEMENTATION AND MANAGEMENT

Today, our prototype model has been field-tested and is ready for implementation. Although the implementation and management of the site are issues beyond the scope of this thesis, we feel it is important to comment about these two important program elements. The original manual survey process was administered once to a limited number of squadrons to obtain a representative cross-section of naval aviation. This process returned a one-time snapshot of the culture of naval aviation safety, for that given time. Our automated ACSA project expands upon the potential scope and frequency of this evaluation process. It is well within the operational capabilities of this web site to receive inputs from one or every Internet-capable aviation squadron in the Navy and Marine Corps.

For example, individual commanding officers could evaluate their squadron either before or after cruise, a Type Commander could evaluate every subordinate command before or after a significant policy change, or all Navy squadrons could be evaluated on a regular basis. By design,

the web site does not restrict the frequency or sample size of survey submissions. The number of clients, however, determines the amount of preparation and administration required by the ACSA program administrator. Although the process is automated, the administrator should monitor server operation as surveys are being submitted, review web site system logs and connections, and respond to a predictable increase in newsgroup information flow. Finally, given the current design, the ACSA program administrator will have to analyze individual squadron data and provide timely squadron feedback. Regardless of the scenario, web site implementation and management will be a crucial facet of the program's overall success and its adoption as a useful tool by squadron commanders.

LIST OF REFERENCES

1.   Ciavarelli, Anthony, and Figlock, Robert, *"The Center for Risk Mitigation Kickoff Conference: Organizational Factors in Naval Aviation Accidents"*, June 1997.

2.   Hamilton, Graham, R., and M. Fisher. **JDBC Database Access with Java, A Tutorial and Annotated Reference.** Reading, Massachusetts: Addison-Wesley, 1997.

3.   JavaSoft, *JDBC Drivers*, Sun Microsystems, http://java.sun.com/products/jdbc/jdbc.drivers.html, September 1997.

4.   Hermann, Eric. **Teach Yourself CGI Programming with PERL 5 in a Week.** Indianapolis, Indiana: Sams.net Publishing, 1997.

5.   Shepherd, Eric. *"Installing Perl 5 on Windows NT to run with Internet Information Server,"* April 20, 1998.

6.   Support Online from Microsoft Technical Support. *"Configuring and Testing a PERL Script with IIS"*, July 16, 1998, http://support.microsoft.com/support/kb/articles/Q150/6/29.asp

7.   http://webreview.com/97/10/10/feature/main.html

8.   http://www.javasoft.com/products/java-server/servlets/index.hmtl

9.   Hettihews, Sanjaya and Kelly Held. **Teach Yourself Active Server Pages in 14 Days.** Indianapolis, Indiana: Sams.net Publishing, 1997.

10. "Architecture of the Servlet Package", http://www.javasoft.com/docs/books/tutorial/servlets/overview/architecture.html

11. From the Servlet Sofware Development Kit (SDK) source file *HttpServlet.java*.

12. "The servletrunner Utility", http://www.javasoft.com/docs/books/tutorial/servlets/servletrunner/index.html

13. http://premium.microsoft.com/msdn/downloads/chiliasp/default.htm

14. http://premium.microsoft.com/msdn/library/backgrnd/html/msdn_actplat.htm

15. http://www.activeserverpages.com/overview

16. http://www.activeserverpages.com/learn/sessionwhat.asp

17. http://www.activeserverpages.com/learn/sessionwhat.asp

18. http://www.aspdeveloper.net/iasdocs/aspdocs/ref/obj/introbj_1.htm

19. http://www.microsoft.com/iis/support/iishelp/htm/asp/iiwacomb.htm

20. http://www.microsoft.com/accessdev/articles/movs202.htm

21. http://www.microsoft.com/accessdev/articles/movs202.htm

22. http://www.microsoft.com/data/archb.htm

23. http://www.microsoft.com/accessdev/articles/movs202.htm

24. http://www.microsoft.com/data/ado/adotechinfo/adovsjdbc.htm

25. Jennings, Roger, **Platinum Edition Using Access 97.** Indianapolis,
          Indiana: Que, 1996.

26. Strebe, Matthew and Charles Perkins. **NT Server 4.0
          Study Guide.** Alameda, California: Sybex, 1998.

27. http://premium.microsoft.com/msdn/library/sdkdoc/certsrv/pconcept_3msy.htm

28. Stewart, James and Ramesh Chandak. **MCSE IIS 4.0 Exam Prep.**
          Scottsdale, Arizona: The Coriolis Group, 1998.

29. http://www.w3.org

30. http://scholar.lib.vt.edu

RTTUZYUW RUHPSGG9842 0890945-UUUU--RUWFNAA.
ZNR UUUUU
R 300944Z MAR 97 ZYB PSN 077820Q24
FM CINCPACFLT PEARL HARBOR HI//N00//
TO ALPACFLT
ALLANTFLT
INFO RUENAAA/ASSTSECNAV RDA WASHINGTON DC//C4I//
RUENAAA/CNO WASHINGTON DC//N00/N09/N095/N2/N4/N41/N43/N46/N6/N6B/
    N8/N80/N85/N86/N87/N88//
RUCBCLF/CINCLANTFLT NORFOLK VA//N00/N6//
RUCBACM/CINCUSACOM NORFOLK VA//J00/J6//
RHHMUNA/USCINCPAC HONOLULU HI//J00/J6//
RHDLCNE/CINCUSNAVEUR LONDON UK//00/N6//
RULSSEA/COMNAVSEASYSCOM WASHINGTON DC//N00/N08/PMS335/PMS3
RUENMED/BUMED WASHINGTON DC//N00//
RHRMDAB/RUCJNAV/COMUSNAVCENT//N00/N6//
RUCTPOA/CNET PENSACOLA FL//N00//
RUEACNP/BUPERS WASHINGTON DC//N00//
RUHEHMS/COMMARFORPAC//CG/G6//
RUCKMAA/COMMARFORLANT//CG/G6//
RULSSPA/COMSPAWARSYSCOM WASHINGTON DC//N00/N05/PMW171/PMW176//
RUWFADO/NAVSTKAIRWARCEN FALLON NV//N00//
RULKSDF/COMNAVSECGRU FT GEORGE G MEADE MD//N00//
RULSAMX/COMNAVSUPSYSCOM MECHANICSBURG PA//N00//
RUWFAFK/COMNAVSPECWARCOM CORONADO CA//N00//
RULSADG/NRL WASHINGTON DC//N00//
RULSWCB/COMNAVCOMTELCOM WASHINGTON DC//N00/N3//
RUCOSAO/NAVMASSO CHESAPEAKE VA//N00/N6//
RUWFOAA/NCCOSC RDTE DIV SAN DIEGO CA//N433//
RHHMHAH/CINCPACFLT PEARL HARBOR HI//N00//
BT
UNCLAS //N05230//
ALPACFLT 008/97
MSGID/GENADMIN/CINCPACFLT/008//
SUBJ/INFORMATION TECHNOLOGY FOR THE 21ST CENTURY//
POC/M.R. SCOTT/CDR N6/CINCPACFLT/-/TEL: 808 471-8637//
POC/D.A. STRAUB/CDR N6/CINCLANTFLT/-/TEL: 757 322-5863//
RMKS/1. THIS IS THE FIRST IN A SERIES OF JOINT CINCPACFLT AND
CINCLANTFLT MESSAGES CONCERNING THE DEVELOPMENT AND IMPLEMENTATION
OFIT-21. THIS MESSAGE PROVIDES IT-21 HARDWARE/SOFTWARE
IMPLEMENTATION STANDARDS FOR PROGRAMS INSTALLING INFORMATION
SYSTEMS ON FLEET UNITS/BASES AND PROVIDES THE FLEET WITH GUIDANCE
ON MAINTAINING EXISTING INFORMATION SYSTEMS UNTIL INSTALLATION OF
IT-21 PRODUCTS. THE IT-21 IMPLEMENTATION STANDARDS OUTLINED BELOW
ARE PROMULGATED IN ADVANCE OF DON-WIDE GUIDANCE FROM THE DON CHIEF
INFORMATION OFFICER (CIO). THE DON CIO WILL PROMULGATE DON-WIDE
STANDARDS FOLLOWING NEGOTIATION OF ENTERPRISE-WIDE NETWORK
OPERATING SYSTEMS AND APPLICATIONS.
2. BACKGROUND: INFORMATION SUPERIORITY IS THE FOUNDATION OF
JOINT VISION 2010 BATTLEFIELD DOMINANCE, AS WELL AS THE
WARFIGHTING VISION FOR EACH SERVICE. NETWORK WARFARE, ROBUST
INFRASTRUCTURE AND INFORMATION DISSEMINATION TO DISPERSED FORCES
ARE KEY ELEMENTS IN ACHIEVING INFORMATION SUPERIORITY. IT-21 IS

A FLEET DRIVEN REPRIORITIZATION OF C4I PROGRAMS OF RECORD TO
ACCELERATE THE TRANSITION TO A PC BASED TACTICAL/TACTICAL SUPPORT
WARFIGHTING NETWORK. THE INACTIVATION OF THE CURRENT DOD MESSAGING
SYSTEM (AUTODIN) BY DEC 99, WITH NO PLANNED NAVY INFRASTRUCTURE
REPLACEMENT, MANDATES THE RAPID IMPLEMENTATION OF THIS WARFIGHTING
NETWORK.
3. COMMERCIAL NETWORK OPERATING SYSTEMS (NOS) AND E-MAIL PRODUCTS
HAVE ACHIEVED FUNCTIONAL PARITY. THE FLEETS CANNOT CONTINUE TO
SUPPORT A MULTITUDE OF DIVERSE OPERATING SYSTEMS AND E-MAIL PRODUCTS
WITH THEIR OWN TRAINING, OPERATIONAL PROCEDURES AND TROUBLESHOOTING
REQUIREMENTS. THE DOD JOINT TECHNICAL ARCHITECTURE (JTA) AND
DEFENSE INFORMATION INFRASTRUCTURE COMMON OPERATING ENVIRONMENT
(DII COE) PROVIDE DOD WITH THE AIS SYSTEM GUIDANCE REQUIRED TO
TAKE THE NAVY INTO THE 21ST CENTURY. THIS CONVERGENCE OF SOLUTIONS,
PROBLEMS AND GUIDANCE PROVIDES THE IMPETUS TO ESTABLISH MINIMUM
NAVY AIS STANDARDS AT THIS TIME. IMPLEMENTATION OF THIS POLICY
REQUIRES ALL NON-STANDARD NOS AND E-MAIL PRODUCTS BE REPLACED NLT
DEC 99.
   A. WINDOWS NT SERVER 4.0 IS THE STANDARD FLEET NOS. IT WILL
SOON BE FOLLOWED BY WINDOWS NT 5.0. WINDOWS NT SERVER 4.0 IS DII
COE COMPLIANT.
   B. MS EXCHANGE IS DESIGNATED AS THE STANDARD E-MAIL SOLUTION
FOR BOTH FLEETS TO ENSURE AN INTEROPERABLE SECURE MESSAGING SYSTEM
IS OPERATIONAL PRIOR TO AUTODIN INACTIVATION NLT DEC 99.
   C. MS OFFICE 97 IS DESIGNATED AS THE STANDARD FLEET OFFICE
SOFTWARE.
   D. EXPENDITURE OF OPERATING FUNDS TO MAINTAIN EXISTING IT-21
NONCOMPLIANT NOS AND APPLICATIONS SHALL BE THE ABSOLUTE MINIMUM
NECESSARY TO MEET OPERATING REQUIREMENTS UNTIL IT-21 NOS/SOFTWARE
IS INSTALLED EVEN IF TEMPORARY LAN DEGRADATION OCCURS. SOFTWARE
REQUIREMENTS DRIVE HARDWARE STANDARDS. HARDWARE AND SOFTWARE
PURCHASED TODAY MUST BE CAPABLE OF MEETING MISSION REQUIREMENTS
THROUGH THE YEAR 2000.
4. CINCPACFLT AND CINCLANTFLT ARE ACTIVELY WORKING WITH OPNAV ON
IT-21 FUNDING AND IMPLEMENTATION PLANS. IN GENERAL, AFLOAT IT-21
IMPLEMENTATION WILL BE LINKED TO DEPLOYING BATTLEGROUPS AND
ASHORE IT-21 WILL BE IMPLEMENTED IN A PHASED APPROACH. SPECIFIC
IMPLEMENTATION SCHEDULES WILL BE PROMULGATED AT A LATER DATE.
CINCPACFLT AND CINCLANTFLT ARE TRANSITIONING TO WINDOWS NT 4.0,
MS EXCHANGE AND MICROSOFT OFFICE 97. THIS ENVIRONMENT CANNOT BE
OPTIMIZED WITHOUT 32 BIT OPERATING SYSTEMS, HIGH RESOLUTION
DISPLAYS AND MASS STORAGE. ATM BACKBONE LANS WITH AT LEAST 100 MBS
(TCP/IP)TO THE DESKTOP PC WILL BE INSTALLED ON ALL SHIPBOARD LANS,
FLEET HEADQUARTERS (CPF, CLF, TYCOMS, GROUP AND SQUADRON COMMANDS)
AND SHOULD BE INSTALLED IN THOSE SHORE ACTIVITIES THAT SUPPORT
TACTICAL OPERATIONS. THIS WILL THEN ALLOW TRANSITION TO ATM-TO-
THE-DESKTOP PC WHEN THE ATM TECHNOLOGY MATURES.
5. SYSTEM COMMANDS AND PROGRAM MANAGERS:
   A. NTCSS WILL BECOME THE IT-21 PROGRAM OF RECORD FOR
INSTALLATION OF BOTH SECRET AND UNCLASSIFIED LANS ONBOARD
COMMISSIONED SHIPS. NTCSS (ATIS/SNAP III) LANS INSTALLED FROM THIS
POINT ON WILL HAVE AN ATM BACKBONE, 100 MBS (FAST ETHERNET) TO THE
DESKTOP PC AND THE HARDWARE/SOFTWARE OUTLINED AT THE END OF THIS
MESSAGE. THE MIGRATION OF NTCSS LANS TO HIGHER CAPACITY LANS WILL
REDUCE THE NUMBER OF PC'S DELIVERED DURING INITIAL INSTALLATION.
THE TRADE-OFF OF QUANTITY FOR FRONT END PC'S IS REQUIRED TO

SUPPORT JV-2010 AND AUTODIN INACTIVATION.

B. SPAWAR IS WORKING WITH NAVSEA TO ENSURE THAT LANS INSTALLED DURING NEW CONSTRUCTION MEET THE IT-21 REQUIREMENTS.

C. APPLICATION PROGRAM MANAGERS SUCH AS JMCIS, NSIPS, TAMPS, AND GCSS SHOULD MIGRATE CURRENT APPLICATIONS TO THE DII COE WITH AN IMMEDIATE OBJECTIVE OF OBTAINING PC WORKSTATION ACCESS TO ALL APPLICATION DATA ON AN ENTERPRISE LAN.

D. PROGRAMS INSTALLING INFORMATION SYSTEMS (NEWNET, SMARTLINK, SMARTBASE, TELEMEDICINE, ETC.) MUST INSTALL COMPONENTS IN FLEET ACTIVITIES THAT MEET IT-21 STANDARDS AND PROVIDE INTEROPERABILITY THROUGHOUT THE WARFIGHTING NETWORK.

6. TYCOMS AND THIRD ECHELON COMMANDS SHALL ENSURE THAT:

A. SHIPS AND ACTIVITIES INSTALLING NEW LANS, UNDERGOING SIGNIFICANT LAN UPGRADES OR THOSE ACTIVITIES WITH STAND ALONE PC'S SHALL INSTALL IT-21 HARDWARE AND SOFTWARE. NEW OR REPLACEMENT SHIPBOARD AND SHORE BASED TACTICAL LANS SHOULD HAVE AN ATM BACKBONE WITH AT LEAST 100 MBS (FAST ETHERNET) TO THE PC. ·

B. SHIPS AND ACTIVITIES WITH EXISTING LANS, WHICH REQUIRE REPLACEMENT OF UNSERVICEABLE HARDWARE, SHORT OF A FULL NETWORK UPGRADE, SHALL INSTALL HARDWARE WHICH MEETS IT-21 STANDARDS. THE NEW EQUIPMENT MAY NOT BE COMPATIBLE WITH THE EXISTING LAN HARDWARE. CINCPACFLT AND CINCLANTFLT BELIEVE THAT ALL AUTOMATED INFORMATION SYSTEMS (AIS) PROCURED MUST BE COMPATIBLE WITH THE IT-21 LAN STANDARDS EVEN IF TEMPORARY LAN DEGRADATION OCCURS. THERE IS ONLY SUFFICIENT FUNDING TO DO IT RIGHT THE FIRST TIME.

7. THE IT-21 STANDARDS BELOW REPRESENT FRONT END MARKET TECHNOLOGY, ARE DYNAMIC IN NATURE, AND WILL CONTINUE TO BE CLOSELY LINKED TO COMMERCIAL TRENDS. THE STANDARDS LISTED BELOW ARE INTENDED TO BE MINIMUM STANDARDS AND WILL BE UPDATED PERIODICALLY.

A. IT-21 LAN:

(1) AFLOAT LAN·STANDARDS - ATM FIBER BACKBONE, 100 MBPS (FAST ETHERNET) TO THE PC.

(2) ASHORE TACTICAL AND HEADQUARTERS COMMAND CENTER STANDARD (CPF, CLF, TYCOMS, GROUP AND SQUADRON COMMANDS) - ATM BACKBONE, 100 MBPS (FAST ETHERNET) TO THE PC.

(3) ASHORE TACTICAL SUPPORT COMMAND STANDARDS·(BASES) - ATM BACKBONE, 100 MBPS (FAST ETHERNET) TO THE PC.

(4) METROPOLITAN AREA NETWORKS (MAN) SHOULD BE CAPABLE OF SUPPORTING AT LEAST OC-3 (155MBS).

B. IT-21 SOFTWARE:
- WINDOWS NT 4.0/5.0 WORKSTATION
- MS OFFICE 97 PROFESSIONAL (WORD 97, POWERPOINT 97, EXCEL 97, S ACCESS 97)
- IBM ANTI VIRUS (NAVY LICENSE, AVAIL FROM NAVCIRT)
- MS BACK OFFICE CLIENT
- MS OUTLOOK 97
- MS EXCHANGE 5.0
- MS IMAGE COMPOSER

C. IT-21 DATABASES. RELATIONAL DATABASES THAT CAN SUPPORT WEB TECHNOLOGY IAW THE COE (ORACLE, SYBASE, SQL SERVER, ACCESS, ETC.) WILL BE USED TO SUPPORT DATA REQUIREMENTS AND APPLICATION DEVELOPMENT. ALL PROCESS ENGINEERING INITIATIVES THAT RESULT IN DESIGN/REDESIGN OF A DATA COLLECTION/CAPTURE SYSTEM MUST USE COE COMPLIANT RELATIONAL DATABASE MANAGEMENT SYSTEMS (RDBMS) SOFTWARE. THIS REQUIREMENT IS PROVIDED TO ENSURE RDBMS INITIATIVES USE COTS APPLICATION SOFTWARE. FOR ADDITIONAL INFORMATION ON RELATIONAL

DATABASES CONTACT CDR SANDY BUCKLES, CPF N67, COMM/DSN (808) 474-6384, NIPRNET U67@CPF-EMH.CPF.NAVY.MIL.

D. MINIMUM IT-21 PC CAPABILITIES: CPF CAN CURRENTLY PURCHASE THE IT-21 STANDARD PC WITH SOFTWARE FOR $3250.00 - $3579.00 - SEE PARA 7(H) AND 7(I).
- 200 MHZ PENTIUM PRO CPU
- 64 MB EDO RAM
- 3.0 GB HARD DRIVE
- 3.5 INCH FLOPPY DISK DRIVE
- 8X IDE CD-ROM
- DUAL PCMCIA/PC CARD READER
- PCI VIDEO W/2MB RAM
- 17 INCH MONITOR (1280 X 1024)
- POINTING DEVICE (TRACKBALL OR MOUSE)
- SOUNDBLASTER (COMPATIBLE) AUDIO CARD WITH SPEAKERS KEYBOARD
- CPU COMPATIBLE 100 MBPS FAST ETHERNET NIC

E. STANDARD IT-21 LAPTOP WORKSTATION: APPROXIMATELY $5300 - SEE PARA 7(H).
- 150 MHZ PENTIUM
- 32 MB EDO RAM
- 12.1 IN SVGA ACTIVE MATRIX COLOR DISPLAY
- 2.1 GB EIDE HDD
- 6X INTERNAL CD-ROM
- MODEM, PCMCIA SLOTS, NIC CARD
- SMART LITHIUM BATTERY

F. IT-21 NT FILE SERVER FOR DIRECTORY NETWORK SERVICE: APPROXIMATELY $26K - SEE PARA 7(H). THESE ARE MINIMUM SPECIFICATIONS. NEEDS OF THE SPECIFIC NETWORK WILL DICTATE REQUIREMENTS.
- DUAL 166 MHZ PENTIUM CPU
- 512K SECONDARY CACHE MEMORY- 256 MB RAM
- TWO 4 GB SCSI HDD
- ONE 6 GB DAT DRIVE
- ONE 3.5 INCH FLOPPY DISK DRIVE
- 6X SCSI CD-ROM
- DUAL PCMCIA/PC CARD READER
- 2 DPT SCSI III CACHING CONTROLLERS (SMARTCACHE 4)
- PCI VIDEO W/2MB RAM
- 17 INCH MONITOR (1280 X 1024)
- POINTING DEVICE (TRACKBALL OR MOUSE)
- KEYBOARD
- TWO CABLETRON CPU COMPATIBLE ATM NIC CARDS
- ANTEC DUAL POWER SUPPLY CASE (HOT SWAPPABLE)

G. IT-21 FILE SERVER/APPLICATION SERVER: APPROXIMATELY $26K - SEE PARA 7(H). SAME AS IT-21 NT FILE SERVER FOR DIRECTORY NETWORK SERVICE WITH THE FOLLOWING CHANGES:
- CHANGE HDD RQRMT TO FIVE 4 GB DRIVES
- CHANGE DAT TO 18 GB.

H. PRICES FOR PC TECHNOLOGY ARE CONSTANTLY CHANGING AND CAN VARY GREATLY DEPENDING ON METHOD OF PROCUREMENT. FOR EXAMPLE, ON 28 MAR 97 AN IT-21 PC PURCHASED DIRECTLY FROM A VENDOR COSTS $3643. GOVERNMENT RATE FOR SMALL PURCHASES (LESS THAN TEN) IS $3579. A BULK PROCUREMENT (MORE THAN SEVENTY-FIVE) COSTS $3250. THE ABOVE PRICES INCLUDE SHIPPING. BULK PROCUREMENTS SHOULD BE MADE THROUGH THE TYPE COMMANDERS WHEN APPROPRIATE. MR. RICK KOOKER, CPF N65, COMM/DSN:(808) 474-5882, NIPRNET: U65@CPF-EMH.CPF.NAVY.MIL IS

AVAILABLE TO ASSIST TYCOMS WITH AIS PROCUREMENT ISSUES.

   I. AS NETWORK COMPUTER TECHNOLOGY EVOLVES SOME COMMANDS MAY BE ABLE TO TRANSITION TO NETWORK COMPUTERS. WHEN CONSIDERING INSTALLATION OF NETWORK COMPUTERS, TOTAL NETWORK COST MUST BE EVALUATED. NETWORK COMPUTERS HAVE NOT MATURED SUFFICIENTLY TO IMPLEMENT THEM IN FLEET PLATFORMS AT THIS TIME.

8. WAIVER REQUESTS FROM THE ABOVE STANDARDS SHOULD BE SUBMITTED DIRECTLY TO THE RESPECTIVE CPF/CLF N6. POINTS OF CONTACT ARE AS FOLLOWS:

   A. CINCLANTFLT: CDR DEBRA STRAUB AT COMM (757) 322-5863, NIPRNET: U6@CLF.NAVY.MIL

   B. CINCPACFLT: CDR MIKE SCOTT AT COMM (808) 474-7860, NIPRNET:U6@CPF-EMH.CPF.NAVY.MIL.//

BT

#9842

NNNN

SOFTWARE LICENSE AGREEMENT
USE OF SOFTWARE ORDERED FROM WEBLOGIC IS PROVIDED ONLY UNDER LICENSE WITH WEBLOGIC. PLEASE READ THE FOLLOWING LICENSE CAREFULLY AND INDICATE YOUR ACCEPTANCE BY USE OF THE ACCEPTANCE BOX. CERTAIN CAPITALIZED TERMS ARE DEFINED IN SECTION 14.

1. LICENSE TERMS

(a) Evaluation Use. The terms of this Section 1(a) are applicable to you if you have registered as an Evaluation Customer. Subject to the terms of this Agreement, WebLogic, Inc. ("WebLogic") grants to you a non-exclusive, non-transferable, license to use the evaluation version of the WebLogic Software (the "Software"), in binary object code form for evaluation and trial purposes only. You may use any third party software products or modules supplied by WebLogic solely with the Software. You may not disclose the results of software performance benchmarks to any third party without WebLogic's prior written consent. This license begins upon downloading and ends thirty (30) days thereafter (the "Evaluation Period"). When the license expires you must stop using the Software. You are forbidden from using the Software for Production Use or offering it for resale under the terms of this Section 1(a). All rights not specifically granted to you herein are retained by WebLogic.

(b) Development Use License. The terms of this Section 1(b) are applicable to you if you have registered as a Development Use Customer. Subject to the terms of this Agreement, WebLogic, Inc. ("WebLogic") grants to you a non-exclusive, non-transferable, fee-bearing license to use the Software in object code form solely for Development Use for the number of users and the number of CPU's and at the Sites, all as specified on Licensee's invoice. You may use any third party software products or modules supplied by WebLogic solely with the Software. You may not disclose the results of software performance benchmarks to any third party without WebLogic's prior written consent. Additional restrictions with respect to third party software, if any, will be delivered to you along with your license key. You are forbidden from using the

Software for Production Use or offering it for resale under the terms of this Section 1(b). If you integrate the Software into an application and intend to resell the resulting integrated application you must contact us to obtain the appropriate distribution license. All rights not specifically granted to you herein are retained by WebLogic.

(c) Production Use License. The terms of this Section 1(c) are applicable to you if you have registered as a Production Use Customer. Subject to the terms of this Agreement, WebLogic Inc. ("WebLogic") grants to you, a non-exclusive, non-transferable, fee-bearing license to use the Software solely for Production Use on the Number of CPU's and at the Sites, all as specified on your invoice. You may use any third party software products or modules

supplied by WebLogic solely with the Software. You may not disclose
the results of software performance benchmarks to any third party
without WebLogic's prior written consent. Additional restrictions,
if any, with respect to third party software will be delivered to
you along with your license key. If you integrate the Software into
an application and intend to resell the resulting integrated
application you must contact us to obtain the appropriate
distribution license. All rights not specifically granted to you
herein are retained by WebLogic.

(d) The terms of this Section 1(d) are applicable to you if you are
a Development Use Customer or Production Use Customer. You may make
a reasonable number of copies of the Software and Documentation
(provided that all copyright and other proprietary notices of
WebLogic and its licensors are reproduced), solely for archival and
emergency back-up purposes, and disaster recovery testing purposes.
If you would like to alter the number of Users, or number or
identity of Sites and Servers indicated on your invoice, you must
contact WebLogic in writing for our approval; you may be subject to
additional fees.

2. LICENSE FEES

2.1 In consideration of the applicable license(s) granted pursuant
to Sections 1(b) or 1(c), you agree to pay WebLogic the applicable
license fee(s) within 30 days of issuance of WebLogic's invoice. All
charges and fees provided for in this Agreement are exclusive of and
do not include any taxes, duties or similar charges imposed by any
government ("Taxes"). You agree to pay or reimburse WebLogic for all
such Taxes (other than taxes on the net income of WebLogic).

2.2 If you are outside the United States, you agree that the amounts
to be remitted to WebLogic are to be the actual amounts due without
withholding taxes or other assessments by authorities anywhere in
the foreign location, which withholding taxes or assessments you
agree to pay. You will promptly furnish WebLogic with certificates
evidencing payment of such amounts.

3. ANNUAL MAINTENANCE AND SUPPORT

You must purchase annual Maintenance and Support from WebLogic in
order to qualify for the Maintenance and Support Services described
in this Section.

3.1 Maintenance and Support Services. Maintenance and Support means
that WebLogic will provide: (a) software upgrades and product
enhancements upon their commercial release, and appropriate
documentation, and (b) technical assistance with respect to the
Software, including (i) clarification of functions and features;
(ii) clarification of documentation; (iii) technical support and
guidance in the operation of the Software; and (iv) Software error
analysis and correction. Major product releases are not covered by
the Maintenance and Support contract. WebLogic will use commercially

reasonable efforts to provide error corrections or work-arounds for the most severe errors as soon as possible and based upon WebLogic classification of the severity of the error. Expanded support or technical assistance is available per request at an additional charge in accordance with WebLogic's then-current policy. Maintenance and Support will be provided only with respect to versions of the Software that, in accordance with WebLogic policy, are then being supported by WebLogic.

3.2 Your Responsibilities. You agree to provide WebLogic with reasonable access to your personnel and equipment, if necessary during normal business hours in order to provide Maintenance and Support. You agree to document and promptly report all errors or malfunctions of the Software to WebLogic.

4. INTEREST

Any amounts not paid when due (including License Fees and Maintenance and Support Fees) will be subject to interest at the lesser of 1.5% per month or the highest amount permissible under applicable law.

5. LIMITED WARRANY/LIMITATIONS ON LIABILITY

5.1 Limited Warranty for Certain Uses. With respect to the Software licensed under Section 1(b) or 1(c) (i.e. Production Use and Development Use Licenses, not Evaluation Use Licenses) WebLogic warrants to you and for your benefit only that the Software you license from WebLogic will perform in substantial accordance with its documentation for a period of thirty (30) days from the date you receive the Software. If during this time period the Software does not perform as warranted, WebLogic shall, at its option, undertake to correct the Software, replace such Software free of charge or, if neither of the foregoing is commercially practicable, terminate this Agreement and refund to you the License Fee. This warranty will not apply if the Software has not been properly installed, used outside the scope of the applicable license, modified, altered or operated in an unwarranted environment.

THE FOREGOING IS WEBLOGIC'S SOLE LIABILITY FOR AND YOUR SOLE AND EXCLUSIVE REMEDY FOR BREACH OF THIS WARRANTY.

5.2 No Other Warranties. EXCEPT AS PROVIDED IN SECTION 5.1, THE SOFTWARE IS PROVIDED "AS-IS"; WEBLOGIC MAKES NO REPRESENTATION OR WARRANTIES, EXPRESS OR IMPLIED WITH RESPECT TO THE SOFTWARE, INCLUDING WITHOUT LIMITATION WARRANTIES OF FITNESS FOR A PARTICULAR USE OR PURPOSE, MERCHANTABILITY, NON-INFRINGEMENT, OR THAT THE SOFTWARE WILL OPERATE WITHOUT INTERRUPTION OR BE ERROR FREE, AND WEBLOGIC HEREBY DISCLAIMS ALL SUCH REPRESENTATIONS AND WARRANTIES. WEBLOGIC MAKES NO REPRESENTATIONS ABOUT THE SUITABILITY OF THE SOFTWARE OR ABOUT THE ACCURACY OF ANY CONTENT OR INFORMATION MADE ACCESSIBLE BY THE SOFTWARE.

5.3 Limitations on Liability.

(a) With Respect to Evaluation Use Customers. EVALUATION USE
SOFTWARE IS PROVIDED GRATUITOUSLY AND, THEREFORE, WEBLOGIC AND ITS
SUPPLIERS SHALL NOT BE LIABLE FOR ANY DAMAGES (INCLUDING DIRECT,
INDIRECT, INCIDENTAL, CONSEQUENTIAL AND SPECIAL DAMAGES) UNDER ANY
THEORY OF LIABILITY (INCLUDING TORT CONTRACT, OR ANY OTHER THEORY)
WHETHER SUFFERED BY YOU OR ANY OTHER USER OF THE SOFTWARE, OR ANY
THIRD PARTY, EVEN IF ADVISED OF THE POSSIBILITY OF SUCH DAMAGES

(b) With Respect to Production Use Customers and Development Use
Customers/Cumulative Liability. WITH RESPECT TO SOFTWARE PROVIDED
UNDER THE TERMS OF A PRODUCTION USE LICENSE OR DEVELOPMENT LICENSE,
WEBLOGIC AND ITS SUPPLIERS SHALL NOT BE LIABLE, UNDER ANY THEORY OF
LIABILITY, INCLUDING TORT (INCLUDING NEGLIGENCE AND STRICT
LIABILITY) CONTRACT, OR OTHERWISE, FOR ANY INDIRECT, INCIDENTAL,
CONSEQUENTIAL OR SPECIAL DAMAGES ARISING OUT OF YOUR USE OF THE
SOFTWARE, OR THE USE OF THE SOFTWARE BY ANY END USER, USER,
DISTRIBUTOR, OR ANY THIRD PARTY, OR IN ANY OTHER WAY ARISING OUT OF
THIS AGREEMENT, EVEN IF ADVISED OF THE POSSIBILITY OF SUCH DAMAGES,
INCLUDING ANY LOSS OF PROFITS, LOSS OF USE, BUSINESS INTERRUPTION,
LOSS OF DATA, OR COST OF COVER. WEBLOGIC'S CUMULATIVE LIABILITY
UNDER THIS AGREEMENT SHALL BE LIMITED TO THE AMOUNT OF LICENSE FEES
PAID BY YOU TO WEBLOGIC HEREUNDER FOR THE SOFTWARE GIVING RISE TO
SUCH LIABILITY.

Some States do not permit disclaimers of certain warranties or
limitations on certain types of liability under certain
circumstances; consequently, some of the foregoing disclaimers and
limitations may not be applicable to you, in whole or in part.

6. SOFTWARE CHANGES

WebLogic reserves the right at any time not to release or to
discontinue release of any Software and to alter prices, features,
specifications, capabilities, functions, licensing terms, release
dates, general availability or other characteristics of the
Software.

7. OWNERSHIP

Title, ownership rights, and all intellectual property rights in and
to the Software and Documentation shall remain the sole and
exclusive property of WebLogic and/or its licensors. You agree to
abide by the copyright law and all other applicable laws of the
United States. You acknowledge that the Software contains valuable
confidential information and trade secrets of WebLogic and/or its
licensors; therefore, you agree not to modify the Software, create
derivative works of, or attempt to decipher, decompile, disassemble
or reverse engineer the Software or assist or encourage any third
party in doing so.

8. INDEMNIFICATION

8.1 WebLogic agrees to indemnify and defend you from and against any
and all claims, actions or proceedings, arising out of any claim
that the Software infringes or violates any U.S. patent, copyright
or trade secret right of any third party; so long as you provide:
(i) prompt written notice to WebLogic of such claim; (ii) cooperate
with WebLogic in the defense and/or settlement thereof, at
WebLogic's expense; and, (iii) allow WebLogic to control the defense
and all related settlement negotiations.

8.2 If it is, or if in the reasonable opinion of WebLogic it is
probable that it will be, determined by a court of competent

jurisdiction that such Software or the sale or use thereof infringes
any patent, copyright, trade secret or trademark of a third party or
if WebLogic is enjoined from distributing such Software, then
WebLogic, at its sole option and expense, may: (i) procure for you
the rights and to the same extent as those granted under this
Agreement; (ii) replace such Software with other software, which
complies with the specification of the Software; or, (iii) modify
the Software, to avoid infringement while continuing to have the
Software meet the Specification or (iv) terminate this Agreement and
return to you your license fee, less an amount commensurate with
your period of use of the Software, as amortized over a three year
period.

8.3 You agree that this Section 8 represents WebLogic's sole
obligation to you and shall be your sole and exclusive remedy
pursuant to this Agreement for intellectual property infringement.

8.4 Limitations. WebLogic shall have no indemnity obligation for
claims of infringement to the extent resulting or alleged to result
from: (i) any combination, operation, or use of any Software with
any programs or equipment not supplied by WebLogic; (ii) any
modification of the Software by a party other than WebLogic; and
(iii) your failure, within a reasonable time frame, to implement any
replacement or modification of Software provided by WebLogic.

9. TERM AND TERMINATION

9.1 This Agreement will take effect upon submission of your
registration form for the Software, and will remain in force until
terminated in accordance with this Agreement. This Agreement may be
terminated by you upon thirty (30) days' prior written notice to
WebLogic or by destroying or returning to us all copies and partial
copies of the Software and Documentation under your control;
provided that no such termination will entitle you to a refund of
any portion of the License Fee. WebLogic may, by written notice to
you, terminate this Agreement immediately if any of the following
events ("Termination Events") occur: (a) you fail to pay any amount
due WebLogic within thirty (30) days after WebLogic gives you
written notice of such nonpayment; or (b) you are in material breach
of any non-monetary provision of this Agreement, which breach, if

137

capable of being cured, is not cured within thirty (30) days after WebLogic gives you written notice thereof.

9.2 Termination of this Agreement will not affect the provisions relating to the payment of amounts due, or provisions limiting or disclaiming WebLogic's liability, which provisions will survive termination of this Agreement.

9.3 Within thirty (30) days after the date of termination or discontinuance of this Agreement for any reason whatsoever, you shall destroy the Software and all copies, in whole or in part, all Documentation relating thereto, and any other Confidential Information in its possession that is in tangible form.

10. UNITED STATES GOVERNMENT RIGHTS

The Software provided under this Agreement is commercial computer software developed exclusively at private expense, and in all respects are proprietary data belonging solely to WebLogic.

10.1 Department of Defense End Users: If the Software is acquired by or on behalf of agencies or units of the Department of Defense (DOD), then, pursuant to DoD FAR Supplement Section 227.7202 and its successors (48 C.F.R. 227.7202) the Government's right to use, reproduce or disclose the Software and any accompanying documentation acquired under this Agreement is subject to the restrictions of this Agreement.

10.2 Civilian Agency End Users: If the Software is acquired by or on behalf of civilian agencies of the United States Government, then, pursuant to FAR Section 12.212 and its successors (48 C.F.R. 12.212), the Government's right to use, reproduce or disclose the Software acquired under this Agreement is subject to the restrictions of this Agreement.

11. LEGAL COMPLIANCE

You may not download or otherwise export or re-export the Software or any underlying information or technology except in full compliance with all United States and other applicable laws and regulations. In particular, but without limitation, none of the Software or underlying information or technology may be downloaded or otherwise exported or re-exported (i) into (or to a national or resident of) Cuba, Iran, Iraq, Libya, North Korea, Syria, or Sudan, or (ii) to anyone on the US Treasury Department's list of Specially Designated Nationals or the US Commerce Department's Table of Deny Orders. By licensing the Software, you are agreeing to the foregoing and you are representing and warranting that you are not located in, under control of, or a national or resident of any such country or on any such list.

12. NON-ASSIGNMENT/BINDING AGREEMENT

Neither this Agreement nor any rights under this Agreement may be assigned or otherwise transferred by you, in whole or in part, whether voluntary or by operation of law without the prior written consent of WebLogic. Subject to the foregoing, this Agreement will be binding upon and will inure to the benefit of the parties and their respective successors and assigns.

13. MISCELLANEOUS

13.1 If any term, condition, or provision in this Agreement is found to be invalid, unlawful or unenforceable to any extent, the remaining terms, conditions and provisions will continue to be valid and enforceable to the fullest extent permitted by law.

13.2 This Agreement (including any addenda hereto signed by both parties) represents the entire agreement of the parties with respect to the subject matter of this Agreement and supersedes all previous communications, representations, understandings and agreements, either oral or written, between the parties with respect to said subject matter.

13.3 This Agreement may not be amended, except in writing, signed by both parties. [No terms, provisions or conditions of any purchase order, acknowledgment or other business form that you may use in connection with the acquisition or licensing of the Software will have any effect on the rights, duties or obligations of the parties under, or otherwise modify, this Agreement, regardless of any failure of WebLogic to object to such terms, provisions or conditions.]

13.4 This License shall be governed by and construed in accordance with the laws of the State of California as applied to agreements made, entered into and performed entirely in California by California residents. You agree that any dispute regarding this License will be heard in the state or federal courts having jurisdiction in San Francisco County, California, and you agree that you shall be subject to the personal jurisdiction of such courts.

13.5 Use of WebLogic, Inc. software constitutes an acceptance of the terms of this agreement. If you do not agree to be bound by these provisions, you are required to destroy all copies of WebLogic software from your equipment immediately.

14. DEFINITIONS

The following terms shall have the following meanings:

"Application" means any use of any of the published Application Programming Interfaces (APIs) documented or referenced in the Documentation, whether such use is from a 3rd-party browser (including but not limited to Netscape Navigator or Microsoft

Internet Explorer), or from another software program designed or modified to use APIs provided with the Software.

"Concurrent Users" means the maximum number of simultaneous Users who may use the Software as set forth on your registration form.

"Development Use" means use of the Software by a Development Use Customer to design, develop and/or test new Applications.

"Documentation" means the technical publications prepared and delivered to you by WebLogic or provided to you on the WebLogic web site relating to use of the Software, such as reference, user, installation, systems administrator and technical guides.

"Production Use" means using the Software in an Application for internal business purposes only. Production Use does not include the right to reproduce the Software for sublicensing, resale, or distribution to any party other than a User, including without limitation, distributing the Software as part of a VAR, OEM, distributor or reseller arrangement.

"Server" means a single computer processor capable of executing the Software.
"Site" means the specific, physical location of a Server, as set forth on your registration form.

"Software" means those items of software, in object code format only, proprietary to WebLogic and/or its suppliers, that you have ordered from WebLogic pursuant to your registration form and under the terms of this Agreement.

Last updated 11/21/97

```java
import java.awt.*;
import java.applet.*;
import java.sql.*;
import java.text.*;
import symantec.itools.awt.StateCheckBox;
import symantec.itools.awt.BorderPanel;
import symantec.itools.awt.shape.Rect;
import symantec.itools.awt.shape.HorizontalLine;
import symantec.itools.awt.shape.VerticalLine;
import symantec.itools.awt.WrappingLabel;
import symantec.itools.awt.Label3D;

public class Results extends Applet
{
    final int PIXEL_FACTOR = 2;
    final int COLUMN_HEIGHT = 200;

    final int COLUMN_WIDTH = 20;
    final int PERCENT = 100;
    final int MIN_SEARCH_SIZE = 10;

    final int DISPLAY_OFFSET = 150;

    final int PERCENT_LABEL_OFFSET = 40;

    final short MIN_SURVEY_NUM = 5;


    java.sql.Connection con;
    java.sql.Statement stmt;
    java.lang.String url = "jdbc:ff-microsoft://spitfire.avsafety.nps.navy.mil:1433/Survey2_DB";

    java.lang.String questions[] = {
        "1. My command conducts adequate reviews and updates of safety standards and operating
        procedures.",
        "2. My command uses an internal audit and hazard reporting system to catch any problems that
        may lead to a mishap.",
        "3. My command has a defined process to set training goals and to review performance.",
        "4. My command closely monitors proficiency and currency standards to ensure aircrew are
        qualified to fly.",
        "5. Command leadership is actively involved in the safety program and the management of safety
        matters.",
        "6. My command has a defined process to effectively manage the high-risk aviator.",
        "7. Human Factors Councils have been successful in identifying aircrew members who pose a risk
        to safety.",
        "8. Human Factors Boards have been successful in reducing chances of an aircraft mishap due to
        the high-risk aviator.",
        "9. My command makes effective use of the flight surgeon to help identify and manage high-risk
        personnel.",
        "10. Command leadership encourages reporting safety discrepancies without the fear of negative
        repercussions.",
        "11. Individuals in my command are willing to report safety violations, unsafe behaviors, or
        hazardous conditions.",
```

141

"12. In my command, peer influence is effective at discouraging violations of standard operating procedures, or safety rules.",
"13. In my command, we believe safety is an integral part of flight operations.",
"14. In my command, anyone, who intentionally violates standard procedures, or safety rules, is swiftly corrected.",
"15. In my command, violations of operating procedures, flying regulations, or general flight discipline are rare.",
"16. Leaders in my command encourage everyone to be safety conscious and to follow the rules.",
"17. In my command, an aviator who persistently violates flight standards and rules will seriously jeopardize his/her career.",
"18. I am not comfortable reporting a safety violation because people in my command would react negatively toward me.",
"19. My command has a reputation for high quality performance.",
"20. My command sets high quality standards and strives to maintain quality control.",
"21. My command closely monitors quality and corrects any deviations from established quality standards.",
"22. Quality standards in my command are clearly stated in formal publications and procedural guides.",
"23. Command leaders permit cutting corners to get a job done.",
"24. Lack of experienced personnel has adversely affected my command's ability to operate safely.",
"25. Safety decisions are made at the proper levels, by the most qualified people in my command.",
"26. Command leaders consider safety issues during the formation and execution of operational and training plans.",
"27. Command leadership has a clear picture of the risks associated with its flight operations.",
"28. My command takes the time to identify and assess risks associated with its flight operations.",
"29. My command does a good job managing risks associated with its flight operations.",
"30. My command has increased the chances of a mishap due to inadequate or incorrect risk assessment.",
"31. I am provided adequate resources (time, staffing, budget, and equipment) to accomplish my job.",
"32. My command provides the right number of flight hours per month for me to fly safely.",
"33. I have adequate time to prepare for my brief and flight.",
"34. Based upon my command's personnel and other assets, the command is over-committed.",
"35. My command has incorporated Operational Risk Management processes in decision making at all levels.",
"36. My supervisor can be relied on to keep his/her word.",
"37. Our command leaders and supervisors can be trusted.",
"38. My command's Safety Officer is highly regarded.",
"39. Our Safety Officer is influential in promoting safety.",
"40. My command is genuinely concerned about safety.",
"41. Command leadership is successful in communicating its safety goals to unit personnel.",
"42. My command provides a positive command climate that promotes safe flight operations.",
"43. Command leadership is actively involved in the safety program and management of safety matters.",
"44. Command leadership sets the example for compliance with flight standards.",
"45. My command ensures that all unit members are responsible and accountable for safe flight operations.",
"46. Command leadership willingly assists in providing advice concerning safety matters.",
"47. Command leadership reacts well to unexpected changes to its plans.",
"48. My command does not hesitate to temporarily restrict from flying individuals who are under high personal stress.",
"49. I am adequately trained to safely conduct all of my flights.",
"50. Moral and motivation in my command are high.",

"51. My command ensures the uniform enforcement of all operating standards among unit members.",
"52. Crew rest standards are enforced at my command.",
"53. In my command, NATOPS tests and check rides are conducted as intended, to candidly assess aircrew qualifications.",
"54. My command provides adequate safety backups to catch possible human errors during high-risk missions.",
"55. Within my command, good communications flow exist up and down the chain of command.",
"56. My command has good two-way communication with external commands.",
"57. Safety education and training are adequate in my command.",
"58. The Safety Department is a well-respected element of my command.",
"59. The Aviation Safety Officer position is a sought-after billet in my command.",
"60. My command's Safety Department keeps me well informed regarding important safety information.",
"61. My command's Aircrew Coordination Training program is helping to improve mission performance and safety."};

```
//declare variables used in calculating the display columns
    int Y_coordDisagree =0, Y_coordNeutral =0, Y_coordAgree =0, Y_coordNotApp =0;

    int rowCount =0, numDisagree =0, numNeutral =0, numAgree =0, numNotApplicable =0, numTotal =0;

    double disagreePixelHeight =0.0, neutralPixelHeight =0.0, agreePixelHeight =0.0,
        notAppPixelHeight =0.0;

    double disagreePercent =0.0, neutralPercent =0.0, agreePercent =0.0, notAppPercent =0.0;

    java.lang.String squad[] = {"HC", "HCS", "HM" ,"HMH", "HMT", "HS", "HSC", "HSL",
                        "VAQ", "VAW", "VF", "VFA", "VMAQ", "VMFA",
                        "VP", "VQ", "VR", "VRC", "VS", "VT", "VX", "Other"};

    java.lang.String service[] = {"USN", "USMC"};

    java.lang.String location[] = {"East Coast", "West Coast", "Other"};

        public void init()
        {


                    // This code is automatically generated by Visual Cafe when you add
                    // components to the visual environment. It instantiates and initializes
                    // the components. To modify the code, only use code syntax that matches
                    // what Visual Cafe can generate, or Visual Cafe may be unable to back
                    // parse your Java file into its visual environment.
                    //{{INIT_CONTROLS
                    setLayout(null);
                    setSize(552,509);
                    borderPanel1 = new symantec.itools.awt.BorderPanel();
                    borderPanel1.setLayout(null);
                    borderPanel1.setBounds(0,0,552,492);
                    borderPanel1.setBackground(new Color(16762880));
                    add(borderPanel1);
                    SubmitQueryButton = new java.awt.Button();
                    SubmitQueryButton.setLabel("Submit Your Query");
                    SubmitQueryButton.setBounds(374,228,120,35);
                    SubmitQueryButton.setFont(new Font("Dialog", Font.BOLD, 12));
```

```
SubmitQueryButton.setForeground(new Color(255));
SubmitQueryButton.setBackground(new Color(12632256));
borderPanel1.add(SubmitQueryButton);
SubmitQueryButton.setCursor(new Cursor(Cursor.HAND_CURSOR));
questionBox = new java.awt.Choice();
questionBox.addItem("Select a survey Question");
questionBox.addItem("Question 1");
questionBox.addItem("Question 2");
questionBox.addItem("Question 3");
questionBox.addItem("Question 4");
questionBox.addItem("Question 5");
questionBox.addItem("Question 6");
questionBox.addItem("Question 7");
questionBox.addItem("Question 8");
questionBox.addItem("Question 9");
questionBox.addItem("Question 10");
questionBox.addItem("Question 11");
questionBox.addItem("Question 12");
questionBox.addItem("Question 13");
questionBox.addItem("Question 14");
questionBox.addItem("Question 15");
questionBox.addItem("Question 16");
questionBox.addItem("Question 17");
questionBox.addItem("Question 18");
questionBox.addItem("Question 19");
questionBox.addItem("Question 20");
questionBox.addItem("Question 21");
questionBox.addItem("Question 22");
questionBox.addItem("Question 23");
questionBox.addItem("Question 24");
questionBox.addItem("Question 25");
questionBox.addItem("Question 26");
questionBox.addItem("Question 27");
questionBox.addItem("Question 28");
questionBox.addItem("Question 29");
questionBox.addItem("Question 30");
questionBox.addItem("Question 31");
questionBox.addItem("Question 32");
questionBox.addItem("Question 33");
questionBox.addItem("Question 34");
questionBox.addItem("Question 35");
questionBox.addItem("Question 36");
questionBox.addItem("Question 37");
questionBox.addItem("Question 38");
questionBox.addItem("Question 39");
questionBox.addItem("Question 40");
questionBox.addItem("Question 41");
questionBox.addItem("Question 42");
questionBox.addItem("Question 43");
questionBox.addItem("Question 44");
questionBox.addItem("Question 45");
questionBox.addItem("Question 46");
questionBox.addItem("Question 47");
questionBox.addItem("Question 48");
questionBox.addItem("Question 49");
questionBox.addItem("Question 50");
```

```java
questionBox.addItem("Question 51");
questionBox.addItem("Question 52");
questionBox.addItem("Question 53");
questionBox.addItem("Question 54");
questionBox.addItem("Question 55");
questionBox.addItem("Question 56");
questionBox.addItem("Question 57");
questionBox.addItem("Question 58");
questionBox.addItem("Question 59");
questionBox.addItem("Question 60");
questionBox.addItem("Question 61");
try {
        questionBox.select(0);
}
catch (IllegalArgumentException e) { }
borderPanel1.add(questionBox);
questionBox.setBounds(350,72,170,40);
questionBox.setFont(new Font("Dialog", Font.BOLD, 12));
questionBox.setForeground(new Color(0));
questionBox.setBackground(new Color(16777215));

//troubleshooting text display area controls. To view values on the applet
//change the textDisplayArea.setVisible(false) to true and recompile.
textDisplayArea = new java.awt.TextArea();
textDisplayArea.setEditable(false);
textDisplayArea.setVisible(false);
textDisplayArea.setBounds(386,288,60,144);
textDisplayArea.setBackground(new Color(16777215));
borderPanel1.add(textDisplayArea);

disagreeColumn = new symantec.itools.awt.shape.Rect();
try {
        disagreeColumn.setFillColor(new java.awt.Color(16711680));
}
catch(java.beans.PropertyVetoException e) { }
try {
        disagreeColumn.setFillMode(true);
}
catch(java.beans.PropertyVetoException e) { }
disagreeColumn.setVisible(false);
disagreeColumn.setBounds(100,150,20,200);
disagreeColumn.setBackground(new Color(16777215));
borderPanel1.add(disagreeColumn);
neutralColumn = new symantec.itools.awt.shape.Rect();
try {
        neutralColumn.setFillMode(true);
}
catch(java.beans.PropertyVetoException e) { }
neutralColumn.setVisible(false);
neutralColumn.setBounds(170,150,20,200);
neutralColumn.setBackground(new Color(0));
borderPanel1.add(neutralColumn);
agreeColumn = new symantec.itools.awt.shape.Rect();
try {
        agreeColumn.setFillColor(new java.awt.Color(255));
}
```

```
catch(java.beans.PropertyVetoException e) { }
try {
        agreeColumn.setFillMode(true);
}
catch(java.beans.PropertyVetoException e) { }
agreeColumn.setVisible(false);
agreeColumn.setBounds(230,150,20,200);
borderPanel1.add(agreeColumn);
notAppColumn = new symantec.itools.awt.shape.Rect();
try {
        notAppColumn.setFillColor(new java.awt.Color(16776960));
}
catch(java.beans.PropertyVetoException e) { }
try {
        notAppColumn.setFillMode(true);
}
catch(java.beans.PropertyVetoException e) { }
notAppColumn.setVisible(false);
notAppColumn.setBounds(295,150,20,200);
borderPanel1.add(notAppColumn);
titleLabel = new java.awt.Label("Command Safety Assessment",Label.CENTER);
titleLabel.setBounds(0,10,530,56);
titleLabel.setFont(new Font("Serif", Font.BOLD, 30));
titleLabel.setForeground(new Color(0));
titleLabel.setBackground(new Color(16762880));
borderPanel1.add(titleLabel);
scaleHorizLine = new symantec.itools.awt.shape.HorizontalLine();
scaleHorizLine.setBounds(50,350,300,2);
borderPanel1.add(scaleHorizLine);
scaleVerticalLine = new symantec.itools.awt.shape.VerticalLine();
scaleVerticalLine.setBounds(50,150,2,200);
borderPanel1.add(scaleVerticalLine);
tenPercentLine = new symantec.itools.awt.shape.HorizontalLine();
tenPercentLine.setBounds(50,330,15,2);
tenPercentLine.setForeground(new Color(16777215));
borderPanel1.add(tenPercentLine);
twentyPercentLine = new symantec.itools.awt.shape.HorizontalLine();
twentyPercentLine.setBounds(50,310,20,2);
borderPanel1.add(twentyPercentLine);
thirtyPercentLine = new symantec.itools.awt.shape.HorizontalLine();
thirtyPercentLine.setBounds(50,290,15,2);
thirtyPercentLine.setForeground(new Color(16777215));
borderPanel1.add(thirtyPercentLine);
fortyPercentLine = new symantec.itools.awt.shape.HorizontalLine();
fortyPercentLine.setBounds(50,270,20,2);
borderPanel1.add(fortyPercentLine);
fiftyPercentLine = new symantec.itools.awt.shape.HorizontalLine();
fiftyPercentLine.setBounds(50,250,15,2);
fiftyPercentLine.setForeground(new Color(16777215));
borderPanel1.add(fiftyPercentLine);
sixtyPercentLine = new symantec.itools.awt.shape.HorizontalLine();
sixtyPercentLine.setBounds(50,230,20,2);
borderPanel1.add(sixtyPercentLine);
seventyPercentLine = new symantec.itools.awt.shape.HorizontalLine();
seventyPercentLine.setBounds(50,210,15,2);
seventyPercentLine.setForeground(new Color(16777215));
```

```
borderPanel1.add(seventyPercentLine);
eithtyPercentLine = new symantec.itools.awt.shape.HorizontalLine();
eithtyPercentLine.setBounds(50,190,20,2);
borderPanel1.add(eithtyPercentLine);
ninetyPercentLine = new symantec.itools.awt.shape.HorizontalLine();
ninetyPercentLine.setBounds(50,170,15,2);
ninetyPercentLine.setForeground(new Color(16777215));
borderPanel1.add(ninetyPercentLine);
hundredPercentLine = new symantec.itools.awt.shape.HorizontalLine();
hundredPercentLine.setBounds(50,150,20,2);
borderPanel1.add(hundredPercentLine);
questionDisplayLabel = new symantec.itools.awt.WrappingLabel();
questionDisplayLabel.setBounds(14,408,492,60);
questionDisplayLabel.setFont(new Font("Dialog", Font.BOLD, 14));
borderPanel1.add(questionDisplayLabel);
questionDisplayLabel.setCursor(new Cursor(Cursor.HAND_CURSOR));
disagreeLabel = new symantec.itools.awt.WrappingLabel();
try {
        disagreeLabel.setText("1 - 3 Disagree");
}
catch(java.beans.PropertyVetoException e) { }
try {
    disagreeLabel.setAlignStyle
        (symantec.itools.awt.WrappingLabel.ALIGN_CENTERED);

}
catch(java.beans.PropertyVetoException e) { }
disagreeLabel.setBounds(78,360,60,36);
disagreeLabel.setFont(new Font("Dialog", Font.BOLD, 12));
borderPanel1.add(disagreeLabel);
neutralLabel = new symantec.itools.awt.WrappingLabel();
try {
        neutralLabel.setText("  4    Neutral");
}
catch(java.beans.PropertyVetoException e) { }
try {
    neutralLabel.setAlignStyle
        (symantec.itools.awt.WrappingLabel.ALIGN_CENTERED);
}
catch(java.beans.PropertyVetoException e) { }
neutralLabel.setBounds(155,360,60,48);
neutralLabel.setFont(new Font("Dialog", Font.BOLD, 12));
borderPanel1.add(neutralLabel);
agreeLabel = new symantec.itools.awt.WrappingLabel();
try {
        agreeLabel.setText("5 - 7  Agree");
}
catch(java.beans.PropertyVetoException e) { }
try {
    agreeLabel.setAlignStyle
        (symantec.itools.awt.WrappingLabel.ALIGN_CENTERED);
}
catch(java.beans.PropertyVetoException e) { }
agreeLabel.setBounds(213,360,60,41);
agreeLabel.setFont(new Font("Dialog", Font.BOLD, 12));
borderPanel1.add(agreeLabel);
```

```java
notAppLabel = new symantec.itools.awt.WrappingLabel();
try {
        notAppLabel.setText("Not     Applicable");
}
catch(java.beans.PropertyVetoException e) { }
try {
    notAppLabel.setAlignStyle
        (symantec.itools.awt.WrappingLabel.ALIGN_CENTERED);
}
catch(java.beans.PropertyVetoException e) { }
notAppLabel.setBounds(269,360,90,36);
notAppLabel.setFont(new Font("Serif", Font.BOLD, 12));
borderPanel1.add(notAppLabel);
zeroPercentLabel = new symantec.itools.awt.WrappingLabel();
try {
        zeroPercentLabel.setText("0 %");
}
catch(java.beans.PropertyVetoException e) { }
try {
    zeroPercentLabel.setAlignStyle
        (symantec.itools.awt.WrappingLabel.ALIGN_CENTERED);
}
catch(java.beans.PropertyVetoException e) { }
zeroPercentLabel.setBounds(14,336,31,24);
zeroPercentLabel.setFont(new Font("Dialog", Font.BOLD, 12));
borderPanel1.add(zeroPercentLabel);
twentyPercentLabel = new symantec.itools.awt.WrappingLabel();
try {
        twentyPercentLabel.setText("20 %");
}
catch(java.beans.PropertyVetoException e) { }
try {
    twentyPercentLabel.setAlignStyle
        (symantec.itools.awt.WrappingLabel.ALIGN_CENTERED);
}
catch(java.beans.PropertyVetoException e) { }
twentyPercentLabel.setBounds(14,302,36,24);
twentyPercentLabel.setFont(new Font("Dialog", Font.BOLD, 12));
borderPanel1.add(twentyPercentLabel);
fortyPercentLabel = new symantec.itools.awt.WrappingLabel();
try {
        fortyPercentLabel.setText("40 %");
}
catch(java.beans.PropertyVetoException e) { }
try {
    fortyPercentLabel.setAlignStyle
        (symantec.itools.awt.WrappingLabel.ALIGN_CENTERED);
}
catch(java.beans.PropertyVetoException e) { }
fortyPercentLabel.setBounds(14,264,36,24);
fortyPercentLabel.setFont(new Font("Dialog", Font.BOLD, 12));
borderPanel1.add(fortyPercentLabel);
sixtyPercentLabel = new symantec.itools.awt.WrappingLabel();
try {
        sixtyPercentLabel.setText("60 %");
}
```

```java
catch(java.beans.PropertyVetoException e) { }
try {
    sixtyPercentLabel.setAlignStyle
        (symantec.itools.awt.WrappingLabel.ALIGN_CENTERED);
}
catch(java.beans.PropertyVetoException e) { }
sixtyPercentLabel.setBounds(14,224,36,24);
sixtyPercentLabel.setFont(new Font("Dialog", Font.BOLD, 12));
borderPanel1.add(sixtyPercentLabel);
eightyPercentLabel = new symantec.itools.awt.WrappingLabel();
try {
        eightyPercentLabel.setText("80 %");
}
catch(java.beans.PropertyVetoException e) { }
try {
    eightyPercentLabel.setAlignStyle
        (symantec.itools.awt.WrappingLabel.ALIGN_CENTERED);
}
catch(java.beans.PropertyVetoException e) { }
eightyPercentLabel.setBounds(14,182,36,24);
eightyPercentLabel.setFont(new Font("Dialog", Font.BOLD, 12));
borderPanel1.add(eightyPercentLabel);
hundredPercentLabel = new symantec.itools.awt.WrappingLabel();
try {
        hundredPercentLabel.setText("100 %");
}
catch(java.beans.PropertyVetoException e) { }
try {
    hundredPercentLabel.setAlignStyle
        (symantec.itools.awt.WrappingLabel.ALIGN_CENTERED);
}
catch(java.beans.PropertyVetoException e) { }
hundredPercentLabel.setBounds(14,144,35,24);
hundredPercentLabel.setFont(new Font("Dialog", Font.BOLD, 12));
borderPanel1.add(hundredPercentLabel);
numSurveysLabel = new symantec.itools.awt.WrappingLabel();
try {
        numSurveysLabel.setText("Number of surveys =");
}
catch(java.beans.PropertyVetoException e) { }
numSurveysLabel.setVisible(false);
numSurveysLabel.setBounds(74,84,228,24);
numSurveysLabel.setFont(new Font("Serif", Font.BOLD, 18));
borderPanel1.add(numSurveysLabel);
disagreePercentLabel = new symantec.itools.awt.Label3D();
disagreePercentLabel.setVisible(false);
disagreePercentLabel.setBounds(90,120,40,15);
disagreePercentLabel.setFont(new Font("Dialog", Font.BOLD, 12));
disagreePercentLabel.setBackground(new Color(16762880));
borderPanel1.add(disagreePercentLabel);
neutralPercentLabel = new symantec.itools.awt.Label3D();
neutralPercentLabel.setVisible(false);
neutralPercentLabel.setBounds(160,120,40,15);
neutralPercentLabel.setFont(new Font("Dialog", Font.BOLD, 12));
neutralPercentLabel.setBackground(new Color(16762880));
borderPanel1.add(neutralPercentLabel);
```

```java
agreePercentLabel = new symantec.itools.awt.Label3D();
agreePercentLabel.setVisible(false);
agreePercentLabel.setBounds(220,120,40,15);
agreePercentLabel.setFont(new Font("Dialog", Font.BOLD, 12));
agreePercentLabel.setBackground(new Color(16762880));
borderPanel1.add(agreePercentLabel);
notAppPercentLabel = new symantec.itools.awt.Label3D();
notAppPercentLabel.setVisible(false);
notAppPercentLabel.setBounds(285,120,40,15);
notAppPercentLabel.setFont(new Font("Dialog", Font.BOLD, 12));
notAppPercentLabel.setBackground(new Color(16762880));
borderPanel1.add(notAppPercentLabel);
squadronCommunityBox = new java.awt.Choice();
squadronCommunityBox.addItem("Select a Community");
squadronCommunityBox.addItem("HC");
squadronCommunityBox.addItem("HCS");
squadronCommunityBox.addItem("HM");
squadronCommunityBox.addItem("HMH");
squadronCommunityBox.addItem("HMT");
squadronCommunityBox.addItem("HS");
squadronCommunityBox.addItem("HSC");
squadronCommunityBox.addItem("HSL");
squadronCommunityBox.addItem("VAQ");
squadronCommunityBox.addItem("VAW");
squadronCommunityBox.addItem("VF");
squadronCommunityBox.addItem("VFA");
squadronCommunityBox.addItem("VMAQ");
squadronCommunityBox.addItem("VMFA");
squadronCommunityBox.addItem("VP");
squadronCommunityBox.addItem("VQ");
squadronCommunityBox.addItem("VR");
squadronCommunityBox.addItem("VRC");
squadronCommunityBox.addItem("VS");
squadronCommunityBox.addItem("VT");
squadronCommunityBox.addItem("VX");
squadronCommunityBox.addItem("Other");
try {
        squadronCommunityBox.select(0);
}
catch (IllegalArgumentException e) { }
borderPanel1.add(squadronCommunityBox);
squadronCommunityBox.setBounds(350,108,170,36);
squadronCommunityBox.setFont(new Font("Dialog", Font.BOLD, 12));
squadronCommunityBox.setBackground(new Color(16777215));
serviceChoiceBox = new java.awt.Choice();
serviceChoiceBox.addItem("Select a Service");
serviceChoiceBox.addItem("USN");
serviceChoiceBox.addItem("USMC");
try {
        serviceChoiceBox.select(0);
}
catch (IllegalArgumentException e) { }
borderPanel1.add(serviceChoiceBox);
serviceChoiceBox.setBounds(350,144,170,36);
serviceChoiceBox.setFont(new Font("Dialog", Font.BOLD, 12));
serviceChoiceBox.setBackground(new Color(16777215));
```

```
                squadronLocationBox = new java.awt.Choice();
                squadronLocationBox.addItem("Select a Location");
                squadronLocationBox.addItem("East Coast");
                squadronLocationBox.addItem("West Coast");
                squadronLocationBox.addItem("Other");
                try {
                        squadronLocationBox.select(0);
                }
                catch (IllegalArgumentException e) { }
                borderPanel1.add(squadronLocationBox);
                squadronLocationBox.setBounds(350,180,170,36);
                squadronLocationBox.setFont(new Font("Dialog", Font.BOLD, 12));
                squadronLocationBox.setBackground(new Color(16777215));
                selectAQuestionLabel = new java.awt.Label
                        ("You must select a question to submit a query.");
                selectAQuestionLabel.setVisible(false);
                selectAQuestionLabel.setBounds(74,252,276,36);
                selectAQuestionLabel.setFont(new Font("Dialog", Font.BOLD, 12));
                borderPanel1.add(selectAQuestionLabel);
                noResultsLabel = new java.awt.Label("Sorry, no records this query.");
                noResultsLabel.setVisible(false);
                noResultsLabel.setBounds(98,228,180,26);
                noResultsLabel.setFont(new Font("Dialog", Font.BOLD, 12));
                borderPanel1.add(noResultsLabel);
                processingQueryLabel = new java.awt.Label
                        ("Processing your query ...",Label.CENTER);
                processingQueryLabel.setVisible(false);
                processingQueryLabel.setBounds(122,192,158,32);
                processingQueryLabel.setFont(new Font("Dialog", Font.BOLD, 12));
                borderPanel1.add(processingQueryLabel);
                //}}


                //{{REGISTER_LISTENERS
                SymMouse aSymMouse = new SymMouse();
                SubmitQueryButton.addMouseListener(aSymMouse);
                SymItem lSymItem = new SymItem();
                questionBox.addItemListener(lSymItem);
                //}}
}//end public void init()

//{{DECLARE_CONTROLS
symantec.itools.awt.BorderPanel borderPanel1;
java.awt.Button SubmitQueryButton;
java.awt.Choice questionBox;
java.awt.TextArea textDisplayArea;
symantec.itools.awt.shape.Rect disagreeColumn;
symantec.itools.awt.shape.Rect neutralColumn;
symantec.itools.awt.shape.Rect agreeColumn;
symantec.itools.awt.shape.Rect notAppColumn;
java.awt.Label titleLabel;
symantec.itools.awt.shape.HorizontalLine scaleHorizLine;
symantec.itools.awt.shape.VerticalLine scaleVerticalLine;
symantec.itools.awt.shape.HorizontalLine tenPercentLine;
symantec.itools.awt.shape.HorizontalLine twentyPercentLine;
symantec.itools.awt.shape.HorizontalLine thirtyPercentLine;
symantec.itools.awt.shape.HorizontalLine fortyPercentLine;
```

```
symantec.itools.awt.shape.HorizontalLine fiftyPercentLine;
symantec.itools.awt.shape.HorizontalLine sixtyPercentLine;
symantec.itools.awt.shape.HorizontalLine seventyPercentLine;
symantec.itools.awt.shape.HorizontalLine eithtyPercentLine;
symantec.itools.awt.shape.HorizontalLine ninetyPercentLine;
symantec.itools.awt.shape.HorizontalLine hundredPercentLine;
symantec.itools.awt.WrappingLabel questionDisplayLabel;
symantec.itools.awt.WrappingLabel disagreeLabel;
symantec.itools.awt.WrappingLabel neutralLabel;
symantec.itools.awt.WrappingLabel agreeLabel;
symantec.itools.awt.WrappingLabel notAppLabel;
symantec.itools.awt.WrappingLabel zeroPercentLabel;
symantec.itools.awt.WrappingLabel twentyPercentLabel;
symantec.itools.awt.WrappingLabel fortyPercentLabel;
symantec.itools.awt.WrappingLabel sixtyPercentLabel;
symantec.itools.awt.WrappingLabel eightyPercentLabel;
symantec.itools.awt.WrappingLabel hundredPercentLabel;
symantec.itools.awt.WrappingLabel numSurveysLabel;
symantec.itools.awt.Label3D disagreePercentLabel;
symantec.itools.awt.Label3D neutralPercentLabel;
symantec.itools.awt.Label3D agreePercentLabel;
symantec.itools.awt.Label3D notAppPercentLabel;
java.awt.Choice squadronCommunityBox;
java.awt.Choice serviceChoiceBox;
java.awt.Choice squadronLocationBox;
java.awt.Label selectAQuestionLabel;
java.awt.Label noResultsLabel;
java.awt.Label processingQueryLabel;

class SymMouse extends java.awt.event.MouseAdapter{

    //------------------------------------------------------------------------------------
    //Function:     mousePressed()
    //Parameters:   event – a
    //Return Type:  None
    //Purpose:      Records mouse click on "submit a query" button.  Calls
    //                  clearDisplay to reset the displays
    //                  in the event of multiple queries.
    //------------------------------------------------------------------------------------
    public void mousePressed(java.awt.event.MouseEvent event)
    {
       Object object = event.getSource();
         if (object == SubmitQueryButton){
           SubmitQueryButton_MousePressed(event);
         }

    }//end mousePressed()


    //------------------------------------------------------------------------------------
    //Function:     clearDisplay()
    //Parameters:   None
    //Return Type:  None
    //Purpose:      Clears all screen displays and display columns.  Function
    //                  called after submitting each query to calculate the
    //                  display data off screen then display once all values are
    //                  calculated.
```

```
//-------------------------------------------------------------------------------------------------------
public void clearDisplay()
{
  disagreeColumn.setVisible(false);
  neutralColumn.setVisible(false);
  agreeColumn.setVisible(false);
  notAppColumn.setVisible(false);
  disagreePercentLabel.setVisible(false);
  neutralPercentLabel.setVisible(false);
  agreePercentLabel.setVisible(false);
  notAppPercentLabel.setVisible(false);
  noResultsLabel.setVisible(false);
  numSurveysLabel.setVisible(false);
  questionDisplayLabel.setVisible(false);

}//end clearDisplay()


//-------------------------------------------------------------------------------------------------------
//Function:      displayData()
//Parameters:   None
//Return Type:  None
//Purpose:      Checks the total number of records for this display,
//              numTotal, against the minimum number of records required
//              to display query data, MIN_SURVEY_NUM. This feature protects
//              individual anonymity at the beginning of the survey
//              process.  If the query returns more than MIN_SURVEY_NUM
//              records then this function sets visible each display
//              object.
//-------------------------------------------------------------------------------------------------------
public void displayData()
{

  //check that the required number of records exist to display data
  if (numTotal < MIN_SURVEY_NUM) {
    clearDisplay();
    //required number did not exist, display no results alert
        noResultsLabel.setVisible(true);
        processingQueryLabel.setVisible(false);
        return;
  }

  //required number of records exists, displaying results
  else {

        clearDisplay();

    try {
      //displays the corresponding survey question
      questionDisplayLabel.setText("Question " + questions[questionBox.getSelectedIndex()-1]);
      questionDisplayLabel.setVisible(true);

      //display the number of surveys used to calculate the display column data
      numSurveysLabel.setText("Number of surveys: " + rowCount);
    } catch(java.beans.PropertyVetoException e) {}

    processingQueryLabel.setVisible(false);
```

153

```java
        numSurveysLabel.setVisible(true);
        noResultsLabel.setVisible(false);
        disagreeColumn.setVisible(true);
        disagreeColumn.setBounds
            (100,Y_coordDisagree,COLUMN_WIDTH,(int)disagreePixelHeight);
        neutralColumn.setVisible(true);
        neutralColumn.setBounds
            (170,Y_coordNeutral,COLUMN_WIDTH,(int)neutralPixelHeight);
        agreeColumn.setVisible(true);
        agreeColumn.setBounds
            (230,Y_coordAgree,COLUMN_WIDTH,(int)agreePixelHeight);
        notAppColumn.setVisible(true);
         notAppColumn.setBounds
            (295,Y_coordNotApp,COLUMN_WIDTH,(int)notAppPixelHeight);
        setPercentLabelVisible();
    }

}//end displayData()


//------------------------------------------------------------------------------------------
//Function:      setPercentLabelVisible()
//Parameters:    None
//Return Type:   None
//Purpose:       Sets the percent display labels to visible and positions
//                    each label above each column by PERCENT_LABEL_OFFSET pixels.
//------------------------------------------------------------------------------------------
public void setPercentLabelVisible()
{

    disagreePercentLabel.setVisible(true);
    neutralPercentLabel.setVisible(true);
    agreePercentLabel.setVisible(true);
    notAppPercentLabel.setVisible(true);

    //calculating position of each display label
    disagreePercentLabel.setBounds(90,Y_coordDisagree - PERCENT_LABEL_OFFSET,45,30);
    neutralPercentLabel.setBounds(160,Y_coordNeutral - PERCENT_LABEL_OFFSET,45,30);
    agreePercentLabel.setBounds(220,Y_coordAgree - PERCENT_LABEL_OFFSET,45,30);
    notAppPercentLabel.setBounds(280,Y_coordNotApp - PERCENT_LABEL_OFFSET,45,30);

}//end setPercentLabelVisible()



//------------------------------------------------------------------------------------------
//Function:      isColumnZero()
//Parameters:    None
//Return Type:   None
//Purpose:       If the percent label for each column is zero then
//                    this function places "0.0" in the percent label.
//------------------------------------------------------------------------------------------

public void isColumnZero()
{
    try {
        NumberFormat nf = NumberFormat.getInstance();
```

```
                nf.setMaximumFractionDigits(2);

            java.lang.String result = nf.format(disagreePercent);
            if (disagreePercent > 0.0) disagreePercentLabel.setText(result);
                else disagreePercentLabel.setText("0.0");
                    result = nf.format(neutralPercent);
            if (neutralPercent > 0.0) neutralPercentLabel.setText(result);
                else neutralPercentLabel.setText("0.0");
                    result = nf.format(agreePercent);
            if (agreePercent > 0.0)agreePercentLabel.setText(result);
                else agreePercentLabel.setText("0.0");
                    result = nf.format(notAppPercent);
            if (notAppPercent > 0.0)notAppPercentLabel.setText(result);
                else notAppPercentLabel.setText("0.0");


    }
            catch(java.beans.PropertyVetoException e) { }
}//end isColumnZero()


//-----------------------------------------------------------------------------------------------------
//Function:       SubmitQueryButton_MousePressed()
//Parameters:   event – mouse button click
//Return Type:  None
//Purpose:       Checks that a question has been selected from the
//               questionBox.  Initiates connection to the database,
//               and sets the visibility of the display columns
//               to true, displaying in positions calculated from
//               the database data.
//-----------------------------------------------------------------------------------------------------
void SubmitQueryButton_MousePressed(java.awt.event.MouseEvent event)
{
  //used in troubleshooting, not visible on normal applet
  textDisplayArea.appendText("\nButton Pressed");

  clearDisplay();
  //checks to verify a question selection was made, if not
  //then clear display and display alert to select a question
  if ((questionBox.getSelectedIndex() == 0) || (questionBox.getSelectedIndex() == -1)){
        selectAQuestionLabel.setVisible(true);
        return;
  }

  //a question was selected, display the results
  else {

    selectAQuestionLabel.setVisible(false);
    processingQueryLabel.setVisible(true);

    makeConnection();
    isColumnZero();
    processQuery();
    closeConnection();
    calculateDisplayColumns();
    isColumnZero();
    displayData();
    displayTroubleShootingData();
```

```java
        }

}//end SubmitQueryButton_MousePressed()


class SymItem implements java.awt.event.ItemListener{
        //--------------------------------------------------------------------------------
        //Function:       itemStateChanged()
        //Parameters:   None
        //Return Type: None
        //Purpose:        Records the selection of a question from the questionBox.
        //--------------------------------------------------------------------------------
        public void itemStateChanged(java.awt.event.ItemEvent event)

        {
                Object object = event.getSource();
                if (object == questionBox)
                        QuestionBox_ItemStateChanged(event);


                }
        }//end itemStateChanged()



        //--------------------------------------------------------------------------------
        //Function:       QuestionBox_ItemStateChanged()
        //Parameters:   event – selection box input
        //Return Type: None
        //Purpose:        Frees any shared resources called by
        //                SubmitQueryButton_MousePressed
        //                when the data has been returned from the database.
        //--------------------------------------------------------------------------------
        void QuestionBox_ItemStateChanged(java.awt.event.ItemEvent event)
        {
                int selection = questionBox.getSelectedIndex();
                textDisplayArea.appendText( selection + " was your selection \n" );
        }//end QuestionBox_ItemStateChanged()



        //--------------------------------------------------------------------------------
        //Function:       makeConnection()
        //Parameters:   None
        //Return Type: None
        //Purpose:        Establishes a connection to the database.
        //--------------------------------------------------------------------------------
        public void makeConnection()
        {
          try {
            Class.forName("connect.microsoft.MicrosoftDriver").newInstance();
            con = DriverManager.getConnection(url, "sa", "");
            stmt = con.createStatement();
            textDisplayArea.appendText("Successful connection!\n");
          }
          catch (Exception e) {
            textDisplayArea.appendText(e.toString());
          }
```

156

```java
}//end makeConnection()


//------------------------------------------------------------------------------------------------------------
//Function:      processQuery()
//Parameters:    None
//Return Type:   None
//Purpose:       Determines the specific query to be passed to the database
//               from the user's selections of: question number, community,
//               service and or location.  Each query selection calls
//               the appropriate query function, e.g. queryService(query).
//               Calls clearDisplay() to clear the display in the event of
//               multiple user queries.
//------------------------------------------------------------------------------------------------------------
public void processQuery()
{
  java.lang.String query = "SELECT Q" ;
  clearDisplay();

  if (   ((serviceChoiceBox.getSelectedIndex() !=0) &&
          (serviceChoiceBox.getSelectedIndex() != -1)) &&
            ((squadronCommunityBox.getSelectedIndex() !=0) &&
             (squadronCommunityBox.getSelectedIndex() != -1)) &&
               ((squadronLocationBox.getSelectedIndex() !=0) &&
                (squadronLocationBox.getSelectedIndex() != -1)) )

        querySquadronCommunityServiceLocation( query );

  else  if ( ((serviceChoiceBox.getSelectedIndex() !=0) &&
              (serviceChoiceBox.getSelectedIndex() != -1)) &&
                ((squadronLocationBox.getSelectedIndex() !=0) &&
                 (squadronLocationBox.getSelectedIndex() != -1)) )

        queryServiceLocation( query );

  else  if ( ((squadronCommunityBox.getSelectedIndex() !=0) &&
              (squadronCommunityBox.getSelectedIndex() != -1)) &&
                (squadronLocationBox.getSelectedIndex() !=0) &&
                (squadronLocationBox.getSelectedIndex() != -1)) )

        querySquadronCommunityLocation( query );


  else  if ( ((serviceChoiceBox.getSelectedIndex() !=0) &&
              (serviceChoiceBox.getSelectedIndex() != -1)) &&
                ((squadronCommunityBox.getSelectedIndex() !=0) &&
                 (squadronCommunityBox.getSelectedIndex() != -1)) )

        querySquadronCommunityService( query );


  else if ( (serviceChoiceBox.getSelectedIndex() !=0) &&
            (serviceChoiceBox.getSelectedIndex() != -1) )

        queryService( query );
```

```java
        else if ( (squadronCommunityBox.getSelectedIndex() !=0) &&
                (squadronCommunityBox.getSelectedIndex() != -1) )

            querySquadronCommunity( query );

        else if ( (squadronLocationBox.getSelectedIndex() !=0) &&
                (squadronLocationBox.getSelectedIndex() != -1) )

            queryLocation( query );
    else
            queryQuestion( query );

}//end processQuery()



//-------------------------------------------------------------------------------------------
//Function:    · querySquadronCommunityServiceLocation()
//Parameters:  query – initial SQL statement
//Return Type: None
//Purpose:     Receives the initial query from processQuery()and
//             concatenates the remaining SQL query to be passed
//             to the database in order to return the user's query.  This
//             specific query returns the question number, community,
//             service and location tuple.  Calls countResponses()
//             to calculate the number of tuples for this query.
//-------------------------------------------------------------------------------------------
public void querySquadronCommunityServiceLocation( java.lang.String query )
{
        textDisplayArea.append
                ("\nQuery -> querySquadronCommunityServiceLocation ") ;
        textDisplayArea.append
                ("\nCommunity ->" + squad[squadronCommunityBox.getSelectedIndex() -1] ) ;
        textDisplayArea.append
                ("\nService ->" + service[serviceChoiceBox.getSelectedIndex() -1] ) ;
        textDisplayArea.append
                ("\nLocation ->" + location[squadronLocationBox.getSelectedIndex() -1] ) ;

        query += questionBox.getSelectedIndex() +
        " FROM Biographical_Data INNER JOIN Questions ON " +
        "\nQuestions.Community = Biographical_Data.Community AND " +
        "\nQuestions.Squadron_Number = Biographical_Data.Squadron_Number AND " +
        "\nQuestions.Survey_Number = Biographical_Data.Survey_Number    WHERE " +

        "\nBiographical_Data.Community = '" +
            squad[squadronCommunityBox.getSelectedIndex() -1] + AND " +
        "\nBiographical_Data.Service = '" +
            service[serviceChoiceBox.getSelectedIndex() -1] +   AND " +
        "\nBiographical_Data.Location = '" +
            location[squadronLocationBox.getSelectedIndex() -1]+ "'";

        textDisplayArea.append("") ;·
        textDisplayArea.append("\nquery ->" + query ) ;
        textDisplayArea.append("") ;
```

```
            countResponses( query );

}//end querySquadronCommunityServiceLocation()


//----------------------------------------------------------------------------------
//Function:       queryServiceLocation()
//Parameters:   query - initial SQL statement
//Return Type:  None
//Purpose:      Receives the initial query from processQuery()and
//              concatenates the remaining SQL query to be passed
//              to the database in order to return the user's query.  This
//              specific query returns the question number,
//              service and location tuple.  Calls countResponses()
//              to calculate the number of tuples for this query.
//----------------------------------------------------------------------------------
public void queryServiceLocation( java.lang.String query )
{
        textDisplayArea.append
                ("\nQuery -> queryServiceLocation ") ;
        textDisplayArea.append
                ("\nService ->" + service[serviceChoiceBox.getSelectedIndex() -1] ) ;
        textDisplayArea.append
                ("\nLocation ->" + location[squadronLocationBox.getSelectedIndex() -1] ) ;

        query += questionBox.getSelectedIndex() +
        " FROM Biographical_Data INNER JOIN Questions ON " +
        "\nQuestions.Community = Biographical_Data.Community AND " +
        "\nQuestions.Squadron_Number = Biographical_Data.Squadron_Number AND " +
        "\nQuestions.Survey_Number = Biographical_Data.Survey_Number WHERE " +
        "\nBiographical_Data.Service = '" + service[serviceChoiceBox.getSelectedIndex() -1] +
        "' AND " + "\nBiographical_Data.Location = '" +
                location[squadronLocationBox.getSelectedIndex() -1]+ "'";

        textDisplayArea.append("") ;
        textDisplayArea.append("\nquery ->" + query ) ;
        textDisplayArea.append("") ;

        countResponses( query );

}//end queryServiceLocation()


//----------------------------------------------------------------------------------
//Function:       querySquadronCommunityLocation()
//Parameters:   query - initial SQL statement
//Return Type:  None
//Purpose:      Receives the initial query from processQuery()and
//              concatenates the remaining SQL query to be passed
//              to the database in order to return the user's query.  This
//              specific query returns the question number, community,
//              and location tuple.  Calls countResponses()
//              to calculate the number of tuples for this query.
//----------------------------------------------------------------------------------
public void querySquadronCommunityLocation( java.lang.String query )
{
```

159

```java
        textDisplayArea.append
                ("\nQuery -> querySquadronCommunityLocation ") ;
        textDisplayArea.append
                ("\nCommunity ->" + squad[squadronCommunityBox.getSelectedIndex() -1] ) ;
        textDisplayArea.append
                ("\nLocation ->" + location[squadronLocationBox.getSelectedIndex() -1] ) ;
        query += questionBox.getSelectedIndex() + "
        FROM Biographical_Data INNER JOIN Questions ON " +
        "\nQuestions.Community = Biographical_Data.Community AND " +
        "\nQuestions.Squadron_Number = Biographical_Data.Squadron_Number AND " +
        "\nQuestions.Survey_Number = Biographical_Data.Survey_Number WHERE " +

        "\nBiographical_Data.Community = '" +
        squad[squadronCommunityBox.getSelectedIndex() -1] + "' AND " +
        "\nBiographical_Data.Location = '" +
        location[squadronLocationBox.getSelectedIndex() -1]+ "'";
        textDisplayArea.append("") ;
        textDisplayArea.append("\nquery ->" + query ) ;
        textDisplayArea.append("") ;
        countResponses( query );

}//end querySquadronCommunityLocation()


//-------------------------------------------------------------------------------------------
//Function:    querySquadronCommunityService()
//Parameters: query - initial SQL statement
//Return Type:  None
//Purpose:     Receives the initial query from processQuery()and
//             concatenates the remaining SQL query to be passed
//             to the database in order to return the user's query.  This
//             specific query returns the question number, community, and
//             service tuple.  Calls countResponses()to calculate
//             the number of tuples for this query.
//-------------------------------------------------------------------------------------------
public void querySquadronCommunityService( java.lang.String query )
{
        textDisplayArea.append
                ("\nQuery -> querySquadronCommunityService ") ;
        textDisplayArea.append
                ("\nCommunity ->" + squad[squadronCommunityBox.getSelectedIndex() -1] ) ;
        textDisplayArea.append
                ("\nService ->" + service[serviceChoiceBox.getSelectedIndex() -1] ) ;

        query += questionBox.getSelectedIndex() + "
        FROM Biographical_Data INNER JOIN Questions ON " +
        "\nQuestions.Community = Biographical_Data.Community AND " +
        "\nQuestions.Squadron_Number = Biographical_Data.Squadron_Number AND " +
        "\nQuestions.Survey_Number = Biographical_Data.Survey_Number WHERE " +

        "\nBiographical_Data.Community = '" +
        squad[squadronCommunityBox.getSelectedIndex() -1] + "' AND " +
        "\nBiographical_Data.Service = '" +
        service[serviceChoiceBox.getSelectedIndex() -1] + "'";

        textDisplayArea.append("") ;
```

```
                textDisplayArea.append("\nquery ->" + query ) ;
                textDisplayArea.append("") ;

                countResponses( query );

}//end querySquadronCommunityService()


//------------------------------------------------------------------------------
//Function:    queryService()
//Parameters: query - initial SQL statement
//Return Type: None
//Purpose:     Receives the initial query from processQuery()and
//             concatenates the remaining SQL query to be passed
//             to the database in order to return the user's query.  This
//             specific query returns the question number and
//             service tuple.  Calls countResponses()
//             to calculate the number of tuples for this query.
//------------------------------------------------------------------------------
public void queryService( java.lang.String query )
{
        textDisplayArea.append
                ("\nQuery -> queryService ") ;
        textDisplayArea.append
                ("\nService ->" + service[serviceChoiceBox.getSelectedIndex() -1] ) ;

        query += questionBox.getSelectedIndex() + "
        FROM Biographical_Data INNER JOIN Questions ON " +
        "\nQuestions.Community = Biographical_Data.Community AND " +
        "\nQuestions.Squadron_Number = Biographical_Data.Squadron_Number AND " +
        "\nQuestions.Survey_Number = Biographical_Data.Survey_Number WHERE " +

        "\nBiographical_Data.Service = '" +
        service[serviceChoiceBox.getSelectedIndex() -1] + "'";

        textDisplayArea.append("") ;
        textDisplayArea.append("\nquery ->" + query ) ;
        textDisplayArea.append("") ;

        countResponses( query );

}//end queryService()

//------------------------------------------------------------------------------
//Function:    querySquadronCommunity()
//Parameters: query - initial SQL statement
//Return Type: None
//Purpose:     Receives the initial query from processQuery()and
//             concatenates the remaining SQL query to be passed
//             to the database in order to return the user's query.  This
//             specific query returns the question number and community
//             tuple.  Calls countResponses()to calculate the number of
//             tuples for this query.
//------------------------------------------------------------------------------
public void querySquadronCommunity( java.lang.String query )
{
```

```java
            textDisplayArea.append
                    ("\nQuery -> querySquadronCommunity ") ;
            textDisplayArea.append
            ("\nCommunity ->" + squad[squadronCommunityBox.getSelectedIndex() -1] ) ;

            query +=  questionBox.getSelectedIndex() + "
            FROM Biographical_Data INNER JOIN Questions ON " +
            "\nQuestions.Community = Biographical_Data.Community AND " +
            "\nQuestions.Squadron_Number = Biographical_Data.Squadron_Number AND " +
            "\nQuestions.Survey_Number = Biographical_Data.Survey_Number WHERE " +

            "\nBiographical_Data.Community = '" +
            squad[squadronCommunityBox.getSelectedIndex() -1] + "'";

            textDisplayArea.append("") ;
            textDisplayArea.append("\nquery ->" + query ) ;
            textDisplayArea.append("") ;

    countResponses( query );

}//end querySquadronCommunity()



//-----------------------------------------------------------------------------
//Function:      queryLocation()
//Parameters: query - initial SQL statement
//Return Type:  None
//Purpose:    . Receives the initial query from processQuery()and
//            concatenates the remaining SQL query to be passed
//            to the database in order to return the user's query.  This
//            specific query returns the question number and location
//            tuple.  Calls countResponses()to calculate the number of
//            tuples for this query.
//-----------------------------------------------------------------------------
public void queryLocation( java.lang.String query )
{
            textDisplayArea.append
                    ("\nQuery -> queryLocation ") ;
            textDisplayArea.append
                    ("\nLocation ->" + location[squadronLocationBox.getSelectedIndex() -1] ) ;

            query +=  questionBox.getSelectedIndex() +
            " FROM Biographical_Data INNER JOIN Questions ON " +
            "\nQuestions.Community = Biographical_Data.Community AND " +
            "\nQuestions.Squadron_Number = Biographical_Data.Squadron_Number AND " +
            "\nQuestions.Survey_Number = Biographical_Data.Survey_Number WHERE " +

            "\nBiographical_Data.Location = '" +
            location[squadronLocationBox.getSelectedIndex() -1]+ "'";

            textDisplayArea.append("") ;
            textDisplayArea.append("\nquery ->" + query ) ;
            textDisplayArea.append("") ;

            countResponses( query );
```

```
}//end queryLocation()


//---------------------------------------------------------------------------------
//Function:    queryQuestion()
//Parameters: query - initial SQL statement
//Return Type: None
//Purpose:     Receives the initial query from processQuery()and
//             concatenates the remaining SQL query to be passed
//             to the database in order to return the user's query.  This
//             specific query returns the question number tuple.
//             Calls countResponses()to calculate the number of
//             tuples for this query.
//---------------------------------------------------------------------------------
public void queryQuestion( java.lang.String query )
{
        textDisplayArea.append("\n Query -> queryQuestion ") ;

   query += questionBox.getSelectedIndex() + " FROM Questions " ;
        textDisplayArea.append("") ;
         textDisplayArea.append("\nquery ->" + query ) ;
        textDisplayArea.append("") ;

        countResponses( query );

}//end queryQuestion()


//---------------------------------------------------------------------------------
//Function:    countResponses()
//Parameters:   query – the completed SQL query, ready to be executed
//Return Type: None
//Purpose:     Receives and executes the specific database query. Counts
//             each individual result tuple recording the number of
//             records in each category.  Counts the total number
//             of records received in numTotal.
//---------------------------------------------------------------------------------
public void countResponses( java.lang.String query )
{

   try{

      ResultSet rs = stmt.executeQuery( query );

      //initialize variables for multiple queries
      numTotal =0;
      numDisagree =0;
      numNeutral =0;
      numAgree =0;
      numNotApplicable =0;
      rowCount =0;

      while (rs.next()) {
         int x = rs.getInt(1);
         if ((x >= 0 ) && ( x <= 2 ))
            numDisagree++;
```

163

```java
        else if (x == 3)
           numNeutral++;
        else if (x > 3 )
           numAgree++;
        else if (x == -1)
           numNotApplicable++;
        rowCount++;
     }//end while

     numTotal = (numDisagree + numNeutral + numAgree + numNotApplicable);
     rs.close();
   }

   catch (Exception e) {
      textDisplayArea.appendText(e.toString());
   }
}//countResponses()


//--------------------------------------------------------------------------------
//Function:    calculateDisplayColumns()
//Parameters:  None
//Return Type: None
//Purpose:     Each display column is calculated and displayed by pixels.
//             Performs floating point division to calculate the ratio of
//             each column height, then multiplies this value by PERCENT
//             and PIXEL_FACTOR to obtain each display column height
//             in pixels.  Since the (0,0) starting coordinate for each
//             display in the upper right of each monitor the Y_coordinate
//             for each display column is calculated by subtracting the
//             calculated column pixel height and DISPLAY_OFFSET from
//             the full (100%) COLUMN_HEIGTH.
//--------------------------------------------------------------------------------
public void calculateDisplayColumns()
{
   //calculating display percent label values
   disagreePercent = ((float)numDisagree/(float)numTotal) * PERCENT;
   neutralPercent =  ((float)numNeutral/(float)numTotal) * PERCENT;
   agreePercent = ((float)numAgree/(float)numTotal) * PERCENT ;
   notAppPercent =((float)numNotApplicable/(float)numTotal) * PERCENT ;


   //calculating the heigth of each display column in pixels
   disagreePixelHeight =
        ((float)numDisagree/(float)numTotal) * PERCENT * PIXEL_FACTOR  ;
   neutralPixelHeight =
        ((float)numNeutral/(float)numTotal) * PERCENT * PIXEL_FACTOR ;
   agreePixelHeight =
        ((float)numAgree/(float)numTotal) * PERCENT * PIXEL_FACTOR ;
   notAppPixelHeight =
        ((float)numNotApplicable/(float)numTotal) * PERCENT * PIXEL_FACTOR ;


   //calculating the starting pixel y-coordinate for each display column
   Y_coordDisagree = COLUMN_HEIGHT - (int)disagreePixelHeight + DISPLAY_OFFSET;
   Y_coordNeutral = COLUMN_HEIGHT - (int)neutralPixelHeight + DISPLAY_OFFSET;
   Y_coordAgree   = COLUMN_HEIGHT - (int)agreePixelHeight + DISPLAY_OFFSET;
   Y_coordNotApp  = COLUMN_HEIGHT - (int)notAppPixelHeight + DISPLAY_OFFSET;
```

```
}//end calculateDisplayColumns()

//----------------------------------------------------------------------------------------------------
//Function:      displayTroubleshootingData()
//Parameters:  None
//Return Type: None
//Purpose:      For use with the normally hidden display text box.  Used
//              for troubleshooting only.
//----------------------------------------------------------------------------------------------------
public void displayTroubleShootingData()
{
    textDisplayArea.append("\nThe number of disagree ---------> " + numDisagree);
    textDisplayArea.append("\nThe number of neutral ----------> " + numNeutral);
    textDisplayArea.append("\nThe number of agree ------------> " + numAgree);
    textDisplayArea.append("\nThe number of notApplicable ----> " + numNotApplicable);
    textDisplayArea.append("");
    textDisplayArea.append("\nCheck values: rowCount ---------> " + rowCount);

    textDisplayArea.append("");
    textDisplayArea.append("\n disagreePercent-> " + (int)disagreePercent);
    textDisplayArea.append("\n neutralPercent--> " + (int)neutralPercent);
    textDisplayArea.append("\n agreePercent----> " + (int)agreePercent);
    textDisplayArea.append("\n notAppPercent---> " + (int)notAppPercent);

}//displayTroubleShootingData

//----------------------------------------------------------------------------------------------------
//Function:      closeConnection()
//Parameters:   None
//Return Type: None
//Purpose:      Frees any shared resources called by
//              SubmitQueryButton_MousePressed when the data has been
//              returned from the database.
//----------------------------------------------------------------------------------------------------
public void closeConnection()
{
    try{
        con.close();
        stmt.close();
    }
    catch (java.sql.SQLException e){
        textDisplayArea.appendText(e.toString());
    }
}// end closeConnection

}//end class SymItem implements java.awt.event.ItemListener
```

APPENDIX D. SURVEY HYPERTEXT MARKUP LANGUAGE (HTML) SOURCE CODE

A.   A BRIEF HISTORY AND BACKGROUND.

Tim Berners-Lee developed the Hypertext Transport Protocol (HTTP) in late 1990 while he was working at CERN, the European Particle Physics Laboratory located in Geneva, Switzerland. HTTP is a universally accessible communications method that can support a wide range of media types over TCP/IP packet-switching networks.

But it wasn't until 1994, when the National Center for Supercomputing Applications (NCSA) released their Mosaic web browser, that HTML, the markup language used for transferring information via HTTP, began to blossom and enjoy widespread use. As the number of web page authors and vendors began to grow exponentially, the need for an HTML specification maintained by an independent vendor-neutral organization became necessary. In October 1994, the World Wide Web Consortium (W3C) became that organization, and it continues today to lead the World Wide Web to its full potential by developing common protocols that promote its evolution and ensure its interoperability [Ref. 29].

The Consortium attempts to find common specifications for the Web so that private organizations can work in their own fields to exploit and build on top of the global information space which is the web [Ref. 29]. The W3C's goals are threefold:

- Provide a repository of information about the World Wide Web (WWW) for developers and users, especially specifications about the Web. The three specifications that make the Web work are:

Component	Purpose
URLs	A uniform naming scheme for locating resources on the web.
HTTP	A protocol for accessing named resources over the web.
Hypertext	A means of producing web-based documents that allows for easy navigation among resources.

Table D.1. World Wide Web (WWW) components. For more information on these components, see [http://www.w3.org/TR/PR-html40/intro/intro.html].

- Provide sample code implementations to embody and promote standards.

- Provide various prototypes and sample applications to demonstrate the use of new technology [Ref. 29].

The Consortium is a neutral forum, where no member has a greater say
than any other. As of October 1998, it was funded by 287 member
organizations, which join at an annual cost of $50,000 USD (full
membership) or $5,000 USD (affiliate membership). A list of the current
members can be found at [http://www.w3.org/Consortium/Member/List.html].

B.   WHAT IS HTML?

The WWW is clearly a simple and inexpensive means for businesses
to distribute many different forms of digital data, but in order to
effectively and efficiently do so, the material needs to be presented
in a common format. This format, or more properly specification, is
known as HTML. HTML is a Standard Generalized Mark-up Language (SGML)
derivative (i.e. it conforms to International Standard ISO 8879, which
requires a well-defined document structure, a relationship between the
document's parts, and a means of checking the document for validity
against a Document Type Definition (DTD)). It is a non-proprietary
specification that is used to publish hypertext on the WWW. HTML can be
created and processed in a wide range of tools, from simple plain text
editors to sophisticated What You See Is What You Get (WYSIWG)
authoring tools such as Microsoft FrontPage [Ref. 30]:
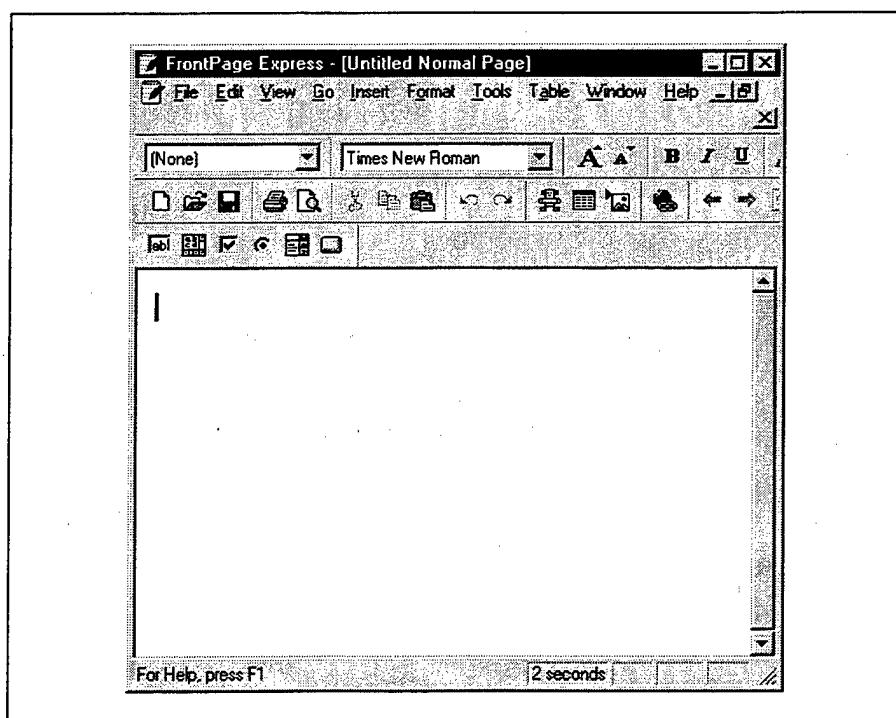
Figure D.1. Microsoft FrontPage (Express version).

168

HTML is a markup language, not a programming language, which
gives authors the ability to:

- Publish documents to the Internet in a platform independent
  format. Documents can include a wide variety of embedded
  elements, such as music, tables, forms, graphics, videos,
  lists, etc... These elements, in turn, have numerous
  attributes that change their presentation style.
- Retrieve information online via hypertext links.
- Link to non-WWW information resources on the Internet.
- Develop applications that can be used for almost any purpose.

Although it was originally intended as a means of encoding document
structures, the HTML specification has evolved to incorporate many new
tags that apply to other text structures, such as headings, paragraphs,
and tables. To use HTML, the author must understand the concept of
markup - what it is and how it is done.

Markup is the act of inserting additional text into a document
that is not usually visible to the reader, and is not part of the
content, but enhances the document in some way. [Ref. 30]. The
additional text that is inserted are tags, which convey formatting
information to the browser. The browser can then decide how to display
the document's content. HTML has two types of markup: *tags* and *character
entities*.

Tags are constructed of a pair of brackets ("<" and ">"), between
which the tag is placed. They are used around segments of text, so
there is usually a companion end tag which is identical to the start
tag except it includes a forward slash [Ref. 30]. Here is an example of
document markup and the use of tags:

```
1:  <HTML>
2:  <HEAD>
3:  <TITLE>My Grocery List</TITLE>
4:  </HEAD>
5:  <BODY>
6:  My Grocery List
7:  <UL>
8:  <LI>Bananas
9:  <LI>Grapes
10: <LI>Apples
11: <LI>Oranges
12: <LI>&#188&nbspI'd rather go to McDonald's&nbsp&#189
13: </UL>
14: </BODY>
```

Figure D.2. Raw HTML code.

When viewed in a web browser (such as Netscape Navigator or Microsoft Internet Explorer), we observe that the markup tags are not visible:
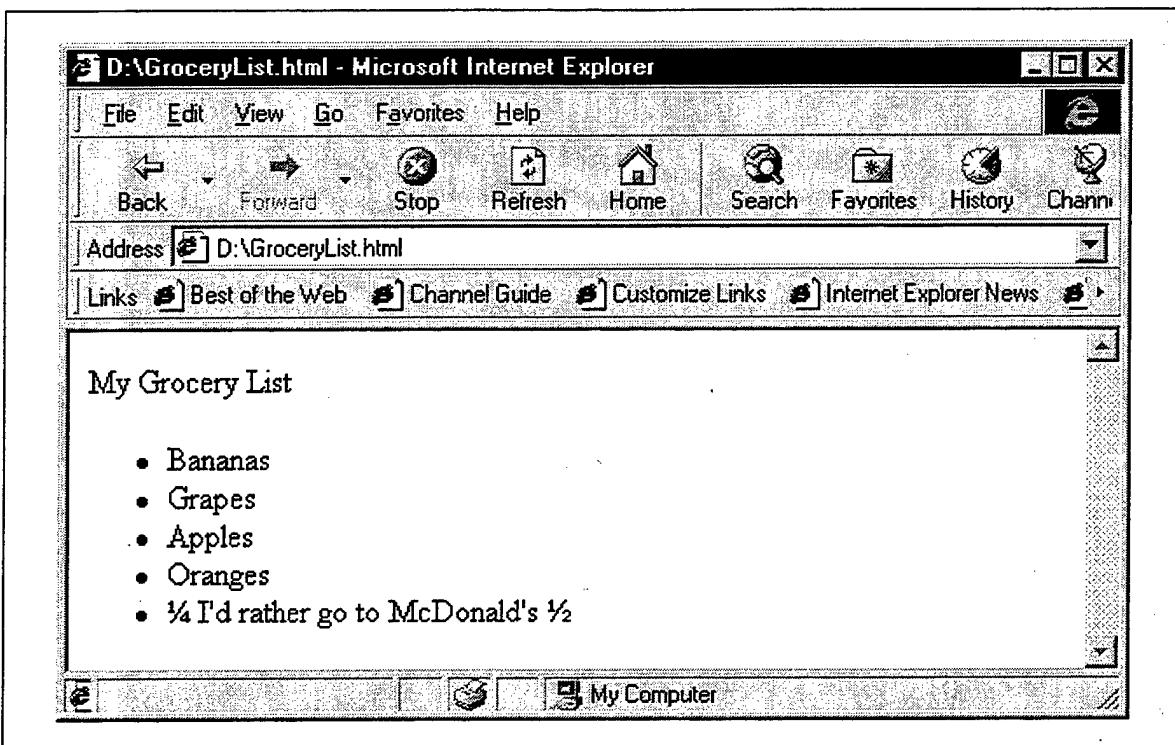
Figure D.3. My Grocery List. The tags are invisible to the client.

When encoding requires the use of special characters (e.g., international characters, mathematical symbols, or Greek letters, SGML entity references may be used. These references may appear in two forms:

- Numeric character references (either decimal or hexadecimal).
  Syntax: "&#D", where D is a decimal number that refers to the Unicode character number D.
- Character entity references.
  Syntax: "&#xH" or "&#XH", where H is a hexadecimal number that refers to the Unicode hexadecimal character number H.

Notice on Line 12 of Figure D.3 the text "&#188" and "&#189". These numeric character references represent the fractions one-quarter and one-half respectively. A complete list of characters can be found at [http://www.w3.org/TR/PR-html40/sgml/entities.html] .

As the HTML specification continues to change, new elements and attributes are added. These additions make authoring easier, and web pages become more robust by allowing a greater set of user interactions. HTML version 4.0 is the most recent specification, and in

170

addition to the text, multimedia, and hyperlink features of previous HTML versions, it supports more multimedia options, scripting languages, style sheets, better printing facilities, and documents that are more accessible to uses with disabilities [Ref. 29].

C.   SURVEY HTML CODE

```
<html>
<head>
<meta http-equiv="Content-Type"
content="text/html; charset=iso-8859-1">
<meta name="keywords"
content="USN, United States Navy, Navy, Aviation, Safety,
Aviation Safety, Naval Aviation">
<meta name="description"
content="The Aviation Safety School automated Command Assessment
Survey Questionnaire. Naval aviation units can use this HTML page, or our Java-enabled
pages to submit survey results.">
<meta name="GENERATOR" content="Microsoft FrontPage Express 2.0">
<title>Aviation Safety Questionnaire</title>
</head>
<body background="images/denim.jpg">
<script language="javascript" src="javascript/checksurvey.js">
</script>
<div align="center"><center>
<table border="0" width="95%">
  <tr>
    <td width="33%"><p align="center"><img
    src="images/ind-deck.gif"
    alt="ind-deck.gif (20333 bytes)" width="175" height="132"></p>
    </td>
    <td width="43%"><p align="center"><font color="#FFFF00"
    size="5"><strong>AVIATION COMMAND SAFETY ASSESSMENT
    (ACSA)<br>
    QUESTIONNAIRE FORM</strong></font></p>
    </td>
    <td width="43%"><p align="center"><img
    src="images/ind-strn.gif"
    alt="ind-strn.gif (16115 bytes)" width="175" height="125"></p>
    </td>
  </tr>
</table>
</center></div>
<form action="http://spitfire.avsafety.nps.navy.mil:8080/servlet/SurveyServlet"
method="post" name="survey" onsubmit="return check_form();">
  <input type="hidden" name="squadron_number" value><input
  type="hidden" name="survey_number" value><blockquote>
    <p align="center"><!--webbot bot="ImageMap" startspan
    rectangle=" (1,5) (52, 38) http://spitfire.avsafety.nps.navy.mil/Help/Help.html##_blank"
    src="images/help.gif" border="0" width="53" height="39" --><MAP
    NAME="FrontPageMap0"><AREA SHAPE="RECT" COORDS="1, 5, 52, 38"
    HREF="http://spitfire.avsafety.nps.navy.mil/Help/Help.html" TARGET="_blank"></MAP><img
    src="images/help.gif" border="0" width="53" height="39" usemap="#FrontPageMap0"><!--webbot
    bot="ImageMap" i-checksum="53268" endspan --></p>
    <p align="center"><font color="#F9CE06" size="4"><strong>PART
```

I.   BIOGRAPHICAL INFORMATION</p>
<p align="left">The
following items are intended to aid our analysis of your
responses. The analyses will be made only on groups
which are sufficiently large to protect individual and
organizational anonymity. NO attempt will be made
to identify individual respondents or their
organizations. Please select the most appropriate
response. </p>
</blockquote>
<div align="center"><center><table border="0" cellspacing="4"
width="64%">
   <tr>
      <td width="36%"><p align="right">Your rank</p>
      </td>
      <td width="64%"><p align="left"><select name="rank"
size="1">
         <option value="X">Select from the following:</option>
         <option value="0">O1 - O3</option>
         <option value="1">O4 - O6</option>
         <option value="2">O6 - O 10</option>
         <option value="3">CWO1 - CWO5</option>
      </select></p>
      </td>
   </tr>
   <tr>
      <td width="36%"><p align="right">Your designation</p>
      </td>
      <td width="64%"><select name="designation" size="1">
         <option value="X">Select from the following:</option>
         <option value="Pilot">Pilot</option>
         <option value="NFO">NFO</option>
         <option value="Other">Other</option>
      </select></td>
   </tr>
   <tr>
      <td width="36%"><p align="right">Your community </p>
      </td>
      <td width="64%"><select name="community" size="1">
         <option value="X">Select from the following:</option>
         <option value="HC">HC</option>
         <option value="HCS">HCS</option>
         <option value="HM">HM</option>
         <option value="HMH">HMH</option>
         <option value="HMT">HMT</option>
         <option value="HS">HS</option>
         <option value="HSC">HSC</option>
         <option value="HSL">HSL</option>
         <option value="VAQ">VAQ</option>
         <option value="VAW">VAW</option>
         <option value="VF">VF</option>
         <option value="VFA">VFA</option>
         <option value="VMAQ">VMAQ</option>

```
      <option value="VMFA">VMFA</option>
      <option value="VP">VP</option>
      <option value="VQ">VQ</option>
      <option value="VR">VR</option>
      <option value="VRC">VRC</option>
      <option value="VS">VS</option>
      <option value="VT">VT</option>
      <option value="VX">VX</option>
      <option value="OTHER">OTHER</option>
    </select></td>
</tr>
<tr>
  <td width="36%"><p align="right"><font
  color="#FFFFFF" size="4">Your total flight hours</font></p>
  </td>
  <td width="64%"><input type="text" size="10"
  name="flight_hours"></td>
</tr>
<tr>
  <td width="36%"><p align="right"><font
  color="#FFFFFF" size="4">Hours in type</font></p>
  </td>
  <td width="64%"><input type="text" size="10"
  name="type_hours"></td>
</tr>
<tr>
  <td width="36%"><p align="right"><font
  color="#FFFFFF" size="4">Are you currently a
  department head?</font></p>
  </td>
  <td width="64%"><select name="department_head"
  size="1">
     <option value="X">Select from the following:</option>
     <option value="Yes">Yes</option>
     <option value="No">No</option>
    </select></td>
</tr>
<tr>
  <td width="36%"><p align="right"><font
  color="#FFFFFF" size="4">Your status</font></p>
  </td>
  <td width="64%"><select name="status" size="1">
     <option value="X">Select from the following</option>
     <option value="Regular">Regular</option>
     <option value="Active Reserve">Active Reserve</option>
     <option value="Drilling Reserve">Drilling Reserve</option>
    </select></td>
</tr>
<tr>
  <td width="36%"><p align="right"><font
  color="#FFFFFF" size="4">Your service</font></p>
  </td>
  <td width="64%"><select name="service" size="1">
     <option value="X">Select from the following:</option>
     <option value="USN">USN</option>
     <option value="USMC">USMC</option>
```

```
                <option value="Other">Other</option>
              </select></td>
          </tr>
          <tr>
            <td width="36%"><p align="right"><font
            color="#FFFFFF" size="4">Your unit's location</font></p>
            </td>
            <td width="64%"><select name="location" size="1">
                <option value="X">Select from the following:</option>
                <option value="East Coast">East Coast</option>
                <option value="West Coast">West Coast</option>
                <option value="Other">Other</option>
              </select></td>
          </tr>
        </table>
</center></div><blockquote>
    <p align="center"><font color="#F9CE06" size="4"><strong>PART
    II.   SURVEY QUESTIONNAIRE</strong></font></p>
    <p align="left"><font color="#F9CE06"><strong>Your honest
    and candid responses to the questionnaire items that
    follow will help your command to make an accurate
    assessment of its safety climate and to estimate the
    success of its ongoing safety program.</strong></font></p>
</blockquote>
<p align="center"><font color="#FFFF00" size="4"
face="Bookman Old Style"><strong>A. PROCESS</strong></font></p>
<div align="center"><center><table border="0" width="100%">
    <tr>
        <td width="11%" bgcolor="#808000"
        style="border-bottom: medium none"><p align="center"><font
        color="#FFFFFF" size="4" face="Times New Roman"><strong>1.</strong></font></p>
        </td>
        <td align="center" width="55%" bgcolor="#808000"
        style="border-bottom: medium none"><font
        color="#FFFFFF" size="4">My command conducts adequate
        reviews and updates of safety standards and operating
        procedures.</font></td>
        <td align="center" width="34%" bgcolor="#808000"
        style="border-bottom: medium none"><p align="center"><select
        name="question1" size="1">
            <option value="X">Select from the following:</option>
            <option value="0">Strongly Disagree</option>
            <option value="1">Moderately Disagree</option>
            <option value="2">Slightly Disagree</option>
            <option value="3">Neutral</option>
            <option value="4">Slightly Agree</option>
            <option value="5">Moderately Agree</option>
            <option value="6">Strongly Agree</option>
            <option value="N">Not Applicable</option>
          </select></p>
        </td>
    </tr>
</table>
</center></div><hr>
<div align="center"><center><table border="0" width="100%">
    <tr>
```

```html
<td width="11%" bgcolor="#C0C0C0"
style="border-bottom: medium none"><p align="center"><font
color="#000000" size="4" face="Times New Roman"><strong>2.</strong></font></p>
</td>
<td align="center" width="55%" bgcolor="#C0C0C0"
style="border-bottom: medium none"><font
color="#000000" size="4">My command uses an internal
audit and hazard reporting system to catch any
problems that may lead to a mishap</font><font
color="#000000">.</font></td>
<td align="center" width="34%" bgcolor="#C0C0C0"
style="border-bottom: medium none"><p align="center"><select
name="question2" size="1">
    <option value="X">Select from the following:</option>
    <option value="0">Strongly Disagree</option>
    <option value="1">Moderately Disagree</option>
    <option value="2">Slightly Disagree</option>
    <option value="3">Neutral</option>
    <option value="4">Slightly Agree</option>
    <option value="5">Moderately Agree</option>
    <option value="6">Strongly Agree</option>
    <option value="-1">Not Applicable</option>
  </select></p>
  </td>
  </tr>
</table>
</center></div><hr>
<div align="center"><center><table border="0" width="100%">
  <tr>
    <td width="11%" bgcolor="#808000"
style="border-bottom: medium none"><p align="center"><font
color="#FFFFFF" size="4" face="Times New Roman"><strong>3.</strong></font></p>
    </td>
    <td align="center" width="55%" bgcolor="#808000"
style="border-bottom: medium none"><font
color="#FFFFFF" size="4">My command has a defined
process to set training goals and to review
performance.</font></td>
    <td align="center" width="34%" bgcolor="#808000"
style="border-bottom: medium none"><p align="center"><select
name="question3" size="1">
    <option value="X">Select from the following:</option>
    <option value="0">Strongly Disagree</option>
    <option value="1">Moderately Disagree</option>
    <option value="2">Slightly Disagree</option>
    <option value="3">Neutral</option>
    <option value="4">Slightly Agree</option>
    <option value="5">Moderately Agree</option>
    <option value="6">Strongly Agree</option>
    <option value="-1">Not Applicable</option>
  </select></p>
  </td>
  </tr>
</table>
</center></div><hr>
<div align="center"><center><table border="0" width="100%">
```

```html
<tr>
    <td width="11%" bgcolor="#C0C0C0"
    style="border-bottom: medium none"><p align="center"><font
    color="#000000" size="4" face="Times New Roman"><strong>4.</strong></font></p>
    </td>
    <td align="center" width="55%" bgcolor="#C0C0C0"
    style="border-bottom: medium none"><font
    color="#000000" size="4">My command closely monitors
    proficiency and currency standards to ensure aircrew
    are qualified to fly.</font></td>
    <td align="center" width="34%" bgcolor="#C0C0C0"
    style="border-bottom: medium none"><p align="center"><select
    name="question4" size="1">
        <option value="X">Select from the following:</option>
        <option value="0">Strongly Disagree</option>
        <option value="1">Moderately Disagree</option>
        <option value="2">Slightly Disagree</option>
        <option value="3">Neutral</option>
        <option value="4">Slightly Agree</option>
        <option value="5">Moderately Agree</option>
        <option value="6">Strongly Agree</option>
        <option value="-1">Not Applicable</option>
    </select></p>
    </td>
</tr>
</table>
</center></div><hr>
<div align="center"><center><table border="0" width="100%">
    <tr>
        <td width="11%" bgcolor="#808000"
        style="border-bottom: medium none"><p align="center"><font
        color="#FFFFFF" size="4" face="Times New Roman"><strong>5.</strong></font></p>
        </td>
        <td align="center" width="55%" bgcolor="#808000"
        style="border-bottom: medium none"><font
        color="#FFFFFF" size="4">Command leadership is
        actively involved in the safety program and
        management of safety matters.</font></td>
        <td align="center" width="34%" bgcolor="#808000"
        style="border-bottom: medium none"><p align="center"><select
        name="question5" size="1">
            <option value="X">Select from the following:</option>
            <option value="0">Strongly Disagree</option>
            <option value="1">Moderately Disagree</option>
            <option value="2">Slightly Disagree</option>
            <option value="3">Neutral</option>
            <option value="4">Slightly Agree</option>
            <option value="5">Moderately Agree</option>
            <option value="6">Strongly Agree</option>
            <option value="-1">Not Applicable</option>
        </select></p>
        </td>
    </tr>
</table>
</center></div><hr>
<div align="center"><center><table border="0" width="100%">
```

```html
<tr>
    <td width="11%" bgcolor="#C0C0C0"
    style="border-bottom: medium none"><p align="center"><font
    color="#000000" size="4" face="Times New Roman"><strong>6.</strong></font></p>
    </td>
    <td align="center" width="55%" bgcolor="#C0C0C0"
    style="border-bottom: medium none"><font size="4">My
    command has a defined process to effectively manage
    the high-risk aviator.</font></td>
    <td align="center" width="34%" bgcolor="#C0C0C0"
    style="border-bottom: medium none"><p align="center"><select
    name="question6" size="1">
        <option value="X">Select from the following:</option>
        <option value="0">Strongly Disagree</option>
        <option value="1">Moderately Disagree</option>
        <option value="2">Slightly Disagree</option>
        <option value="3">Neutral</option>
        <option value="4">Slightly Agree</option>
        <option value="5">Moderately Agree</option>
        <option value="6">Strongly Agree</option>
        <option value="-1">Not Applicable</option>
    </select></p>
    </td>
</tr>
</table>
</center></div><hr>
<div align="center"><center><table border="0" width="100%">
    <tr>
    <td width="11%" bgcolor="#808000"
    style="border-bottom: medium none"><p align="center"><font
    color="#FFFFFF" size="4" face="Times New Roman"><strong>7.</strong></font></p>
    </td>
    <td align="center" width="55%" bgcolor="#808000"
    style="border-bottom: medium none"><font
    color="#FFFFFF" size="4">Human Factors Councils have
    been successful in identifying aircrew members who
    pose a risk to safety.</font></td>
    <td align="center" width="34%" bgcolor="#808000"
    style="border-bottom: medium none"><p align="center"><select
    name="question7" size="1">
        <option value="X">Select from the following:</option>
        <option value="0">Strongly Disagree</option>
        <option value="1">Moderately Disagree</option>
        <option value="2">Slightly Disagree</option>
        <option value="3">Neutral</option>
        <option value="4">Slightly Agree</option>
        <option value="5">Moderately Agree</option>
        <option value="6">Strongly Agree</option>
        <option value="-1">Not Applicable</option>
    </select></p>
    </td>
    </tr>
</table>
</center></div><hr>
<div align="center"><center><table border="0" width="100%">
    <tr>
```

```html
<td width="11%" bgcolor="#C0C0C0"
style="border-bottom: medium none"><p align="center"><font
color="#000000" size="4" face="Times New Roman"><strong>8.</strong></font></p>
</td>
<td align="center" width="55%" bgcolor="#C0C0C0"
style="border-bottom: medium none"><font
color="#000000" size="4">Human Factors Boards have
been successful reducing chances of an aircraft
mishap due to high-risk aviator.</font></td>
<td align="center" width="34%" bgcolor="#C0C0C0"
style="border-bottom: medium none"><p align="center"><select
name="question8" size="1">
    <option value="X">Select from the following:</option>
    <option value="0">Strongly Disagree</option>
    <option value="1">Moderately Disagree</option>
    <option value="2">Slightly Disagree</option>
    <option value="3">Neutral</option>
    <option value="4">Slightly Agree</option>
    <option value="5">Moderately Agree</option>
    <option value="6">Strongly Agree</option>
    <option value="-1">Not Applicable</option>
</select></p>
</td>
</tr>
</table>
</center></div><hr>
<div align="center"><center><table border="0" width="100%">
<tr>
    <td width="11%" bgcolor="#808000"
style="border-bottom: medium none"><p align="center"><font
color="#FFFFFF" size="4" face="Times New Roman"><strong>9.</strong></font></p>
    </td>
    <td align="center" width="55%" bgcolor="#808000"
style="border-bottom: medium none"><font
color="#FFFFFF" size="4">My command makes effective
use of the flight surgeon to help identify and manage
high-risk personnel.</font></td>
    <td align="center" width="34%" bgcolor="#808000"
style="border-bottom: medium none"><p align="center"><select
name="question9" size="1">
    <option value="X">Select from the following:</option>
    <option value="0">Strongly Disagree</option>
    <option value="1">Moderately Disagree</option>
    <option value="2">Slightly Disagree</option>
    <option value="3">Neutral</option>
    <option value="4">Slightly Agree</option>
    <option value="5">Moderately Agree</option>
    <option value="6">Strongly Agree</option>
    <option value="-1">Not Applicable</option>
</select></p>
    </td>
</tr>
</table>
</center></div><p align="center"><font color="#FFFF00"
size="4" face="Bookman Old Style"><strong>B. REWARD SYSTEM
AND SAFETY CULTURE</strong></font></p>
```

```html
<div align="center"><center><table border="0" width="100%">
  <tr>
    <td width="11%" bgcolor="#C0C0C0"
    style="border-bottom: medium none"><p align="center"><font
    color="#000000" size="4" face="Times New Roman"><strong>10.</strong></font></p>
    </td>
    <td align="center" width="55%" bgcolor="#C0C0C0"
    style="border-bottom: medium none"><font
    color="#000000" size="4">Command leadership
    encourages reporting safety discrepancies without the
    fear of negative repercussions.</font></td>
    <td align="center" width="34%" bgcolor="#C0C0C0"
    style="border-bottom: medium none"><p align="center"><select
    name="question10" size="1">
      <option value="X">Select from the following:</option>
      <option value="0">Strongly Disagree</option>
      <option value="1">Moderately Disagree</option>
      <option value="2">Slightly Disagree</option>
      <option value="3">Neutral</option>
      <option value="4">Slightly Agree</option>
      <option value="5">Moderately Agree</option>
      <option value="6">Strongly Agree</option>
      <option value="-1">Not Applicable</option>
    </select></p>
    </td>
  </tr>
</table>
</center></div><hr>
<div align="center"><center><table border="0" width="100%">
  <tr>
    <td width="11%" bgcolor="#808000"
    style="border-bottom: medium none"><p align="center"><font
    color="#FFFFFF" size="4" face="Times New Roman"><strong>11.</strong></font></p>
    </td>
    <td align="center" width="55%" bgcolor="#808000"
    style="border-bottom: medium none"><font
    color="#FFFFFF" size="4">Individuals in my command
    are willing to report safety violations, unsafe
    behaviors or hazardous conditions.</font></td>
    <td align="center" width="34%" bgcolor="#808000"
    style="border-bottom: medium none"><p align="center"><select
    name="question11" size="1">
      <option value="X">Select from the following:</option>
      <option value="0">Strongly Disagree</option>
      <option value="1">Moderately Disagree</option>
      <option value="2">Slightly Disagree</option>
      <option value="3">Neutral</option>
      <option value="4">Slightly Agree</option>
      <option value="5">Moderately Agree</option>
      <option value="6">Strongly Agree</option>
      <option value="-1">Not Applicable</option>
    </select></p>
    </td>
  </tr>
</table>
</center></div><hr>
```

```html
<div align="center"><center><table border="0" width="100%">
  <tr>
    <td width="11%" bgcolor="#C0C0C0"
    style="border-bottom: medium none"><p align="center"><font
    color="#000000" size="4" face="Times New Roman"><strong>12.</strong></font></p>
    </td>
    <td align="center" width="55%" bgcolor="#C0C0C0"
    style="border-bottom: medium none"><font
    color="#000000" size="4">In my command, peer
    influence is effective at discouraging violations of
    standard operating procedures, or safety rules.</font></td>
    <td align="center" width="34%" bgcolor="#C0C0C0"
    style="border-bottom: medium none"><p align="center"><select
    name="question12" size="1">
      <option value="X">Select from the following:</option>
      <option value="0">Strongly Disagree</option>
      <option value="1">Moderately Disagree</option>
      <option value="2">Slightly Disagree</option>
      <option value="3">Neutral</option>
      <option value="4">Slightly Agree</option>
      <option value="5">Moderately Agree</option>
      <option value="6">Strongly Agree</option>
      <option value="-1">Not Applicable</option>
    </select></p>
    </td>
  </tr>
</table>
</center></div><hr>
<div align="center"><center><table border="0" width="100%">
  <tr>
    <td width="11%" bgcolor="#808000"
    style="border-bottom: medium none"><p align="center"><font
    color="#FFFFFF" size="4" face="Times New Roman"><strong>13.</strong></font></p>
    </td>
    <td align="center" width="55%" bgcolor="#808000"
    style="border-bottom: medium none"><font
    color="#FFFFFF" size="4">In my command, we believe
    safety is an integral part of all flight operations.</font></td>
    <td align="center" width="34%" bgcolor="#808000"
    style="border-bottom: medium none"><p align="center"><select
    name="question13" size="1">
      <option value="X">Select from the following:</option>
      <option value="0">Strongly Disagree</option>
      <option value="1">Moderately Disagree</option>
      <option value="2">Slightly Disagree</option>
      <option value="3">Neutral</option>
      <option value="4">Slightly Agree</option>
      <option value="5">Moderately Agree</option>
      <option value="6">Strongly Agree</option>
      <option value="-1">Not Applicable</option>
    </select></p>
    </td>
  </tr>
</table>
</center></div><hr>
<div align="center"><center><table border="0" width="100%">
```

```html
<tr>
    <td width="11%" bgcolor="#C0C0C0"
    style="border-bottom: medium none"><p align="center"><font
    color="#000000" size="4" face="Times New Roman"><strong>14.</strong></font></p>
    </td>
    <td align="center" width="55%" bgcolor="#C0C0C0"
    style="border-bottom: medium none"><font
    color="#000000" size="4">In my command, anyone who
    intentionally violates standard procedures, or safety
    rules, is swiftly corrected.</font></td>
    <td align="center" width="34%" bgcolor="#C0C0C0"
    style="border-bottom: medium none"><p align="center"><select
    name="question14" size="1">
        <option value="X">Select from the following:</option>
        <option value="0">Strongly Disagree</option>
        <option value="1">Moderately Disagree</option>
        <option value="2">Slightly Disagree</option>
        <option value="3">Neutral</option>
        <option value="4">Slightly Agree</option>
        <option value="5">Moderately Agree</option>
        <option value="6">Strongly Agree</option>
        <option value="-1">Not Applicable</option>
    </select></p>
    </td>
  </tr>
</table>
</center></div><hr>
<div align="center"><center><table border="0" width="100%">
    <tr>
    <td width="11%" bgcolor="#808000"
    style="border-bottom: medium none"><p align="center"><font
    color="#FFFFFF" size="4" face="Times New Roman"><strong>15.</strong></font></p>
    </td>
    <td align="center" width="55%" bgcolor="#808000"
    style="border-bottom: medium none"><font
    color="#FFFFFF" size="4">In my command, violations of
    operating procedures, flying regulations, or general
    flight discipline are rare.</font></td>
    <td align="center" width="34%" bgcolor="#808000"
    style="border-bottom: medium none"><p align="center"><select
    name="question15" size="1">
        <option value="X">Select from the following:</option>
        <option value="0">Strongly Disagree</option>
        <option value="1">Moderately Disagree</option>
        <option value="2">Slightly Disagree</option>
        <option value="3">Neutral</option>
        <option value="4">Slightly Agree</option>
        <option value="5">Moderately Agree</option>
        <option value="6">Strongly Agree</option>
        <option value="-1">Not Applicable</option>
    </select></p>
    </td>
  </tr>
</table>
</center></div><hr>
<div align="center"><center><table border="0" width="100%">
```

181

```html
<tr>
    <td width="11%" bgcolor="#C0C0C0"
    style="border-bottom: medium none"><p align="center"><font
    color="#000000" size="4" face="Times New Roman"><strong>16.</strong></font></p>
    </td>
    <td align="center" width="55%" bgcolor="#C0C0C0"
    style="border-bottom: medium none"><font
    color="#000000" size="4">Leaders in my command
    encourage everyone to be safety conscious and to
    follow the rules.</font></td>
    <td align="center" width="34%" bgcolor="#C0C0C0"
    style="border-bottom: medium none"><p align="center"><select
    name="question16" size="1">
        <option value="X">Select from the following:</option>
        <option value="0">Strongly Disagree</option>
        <option value="1">Moderately Disagree</option>
        <option value="2">Slightly Disagree</option>
        <option value="3">Neutral</option>
        <option value="4">Slightly Agree</option>
        <option value="5">Moderately Agree</option>
        <option value="6">Strongly Agree</option>
        <option value="-1">Not Applicable</option>
    </select></p>
    </td>
</tr>
</table>
</center></div><hr>
<div align="center"><center><table border="0" width="100%">
    <tr>
        <td width="11%" bgcolor="#808000"
        style="border-bottom: medium none"><p align="center"><font
        color="#FFFFFF" size="4" face="Times New Roman"><strong>17.</strong></font></p>
        </td>
        <td align="center" width="55%" bgcolor="#808000"
        style="border-bottom: medium none"><font
        color="#FFFFFF" size="4">In this command, an aviator
        who persistently violates flight standards and rules
        will seriously jeopardize his/her career.</font></td>
        <td align="center" width="34%" bgcolor="#808000"
        style="border-bottom: medium none"><p align="center"><select
        name="question17" size="1">
            <option value="X">Select from the following:</option>
            <option value="0">Strongly Disagree</option>
            <option value="1">Moderately Disagree</option>
            <option value="2">Slightly Disagree</option>
            <option value="3">Neutral</option>
            <option value="4">Slightly Agree</option>
            <option value="5">Moderately Agree</option>
            <option value="6">Strongly Agree</option>
            <option value="-1">Not Applicable</option>
        </select></p>
        </td>
    </tr>
</table>
</center></div><hr>
<div align="center"><center><table border="0" width="100%">
```

```html
<tr>
    <td width="11%" bgcolor="#C0C0C0"
    style="border-bottom: medium none"><p align="center"><font
    color="#000000" size="4" face="Times New Roman"><strong>18.</strong></font></p>
    </td>
    <td align="center" width="55%" bgcolor="#C0C0C0"
    style="border-bottom: medium none"><font
    color="#000000" size="4">I am not comfortable
    reporting a safety violation, because people in my
    command would react negatively toward me.</font></td>
    <td align="center" width="34%" bgcolor="#C0C0C0"
    style="border-bottom: medium none"><p align="center"><select
    name="question18" size="1">
        <option value="X">Select from the following:</option>
        <option value="0">Strongly Disagree</option>
        <option value="1">Moderately Disagree</option>
        <option value="2">Slightly Disagree</option>
        <option value="3">Neutral</option>
        <option value="4">Slightly Agree</option>
        <option value="5">Moderately Agree</option>
        <option value="6">Strongly Agree</option>
        <option value="-1">Not Applicable</option>
    </select></p>
    </td>
</tr>
</table>
</center></div><p align="center"><font color="#FFFF00"
size="4" face="Bookman Old Style"><strong>C. QUALITY</strong></font></p>
<div align="center"><center><table border="0" width="100%">
    <tr>
        <td width="11%" bgcolor="#808000"
        style="border-bottom: medium none"><p align="center"><font
        color="#FFFFFF" size="4" face="Times New Roman"><strong>19.</strong></font></p>
        </td>
        <td align="center" width="55%" bgcolor="#808000"
        style="border-bottom: medium none"><font
        color="#FFFFFF" size="4">My command has a reputation
        for high-quality performance.</font></td>
        <td align="center" width="34%" bgcolor="#808000"
        style="border-bottom: medium none"><p align="center"><select
        name="question19" size="1">
            <option value="X">Select from the following:</option>
            <option value="0">Strongly Disagree</option>
            <option value="1">Moderately Disagree</option>
            <option value="2">Slightly Disagree</option>
            <option value="3">Neutral</option>
            <option value="4">Slightly Agree</option>
            <option value="5">Moderately Agree</option>
            <option value="6">Strongly Agree</option>
            <option value="-1">Not Applicable</option>
        </select></p>
        </td>
    </tr>
</table>
</center></div><hr>
<div align="center"><center><table border="0" width="100%">
```

```
<tr>
    <td width="11%" bgcolor="#C0C0C0"
    style="border-bottom: medium none"><p align="center"><font
    color="#000000" size="4" face="Times New Roman"><strong>20.</strong></font></p>
    </td>
    <td align="center" width="55%" bgcolor="#C0C0C0"
    style="border-bottom: medium none"><font
    color="#000000" size="4">My command sets high quality
    standards and strives to maintain quality control.</font></td>
    <td align="center" width="34%" bgcolor="#C0C0C0"
    style="border-bottom: medium none"><p align="center"><select
    name="question20" size="1">
        <option value="X">Select from the following:</option>
        <option value="0">Strongly Disagree</option>
        <option value="1">Moderately Disagree</option>
        <option value="2">Slightly Disagree</option>
        <option value="3">Neutral</option>
        <option value="4">Slightly Agree</option>
        <option value="5">Moderately Agree</option>
        <option value="6">Strongly Agree</option>
        <option value="-1">Not Applicable</option>
    </select></p>
    </td>
</tr>
</table>
</center></div><hr>
<div align="center"><center><table border="0" width="100%">
    <tr>
        <td width="11%" bgcolor="#808000"
        style="border-bottom: medium none"><p align="center"><font
        color="#FFFFFF" size="4" face="Times New Roman"><strong>21.</strong></font></p>
        </td>
        <td align="center" width="55%" bgcolor="#808000"
        style="border-bottom: medium none"><font    .
        color="#FFFFFF" size="4">My command closely monitors
        quality and corrects any deviations from established
        quality standards.</font></td>
        <td align="center" width="34%" bgcolor="#808000"
        style="border-bottom: medium none"><p align="center"><select
        name="question21" size="1">
            <option value="X">Select from the following:</option>
            <option value="0">Strongly Disagree</option>
            <option value="1">Moderately Disagree</option>
            <option value="2">Slightly Disagree</option>
            <option value="3">Neutral</option>
            <option value="4">Slightly Agree</option>
            <option value="5">Moderately Agree</option>
            <option value="6">Strongly Agree</option>
            <option value="-1">Not Applicable</option>
        </select></p>
        </td>
    </tr>
</table>
</center></div><hr>
<div align="center"><center><table border="0" width="100%">
    <tr>
```

```
<td width="11%" bgcolor="#C0C0C0"
style="border-bottom: medium none"><p align="center"><font
color="#000000" size="4" face="Times New Roman"><strong>22.</strong></font></p>
</td>
<td align="center" width="55%" bgcolor="#C0C0C0"
style="border-bottom: medium none"><font
color="#000000" size="4">Quality standards in my
command are clearly stated in formal publications and
procedural guides.</font></td>
<td align="center" width="34%" bgcolor="#C0C0C0"
style="border-bottom: medium none"><p align="center"><select
name="question22" size="1">
    <option value="X">Select from the following:</option>
    <option value="0">Strongly Disagree</option>
    <option value="1">Moderately Disagree</option>
    <option value="2">Slightly Disagree</option>
    <option value="3">Neutral</option>
    <option value="4">Slightly Agree</option>
    <option value="5">Moderately Agree</option>
    <option value="6">Strongly Agree</option>
    <option value="-1">Not Applicable</option>
</select></p>
</td>
</tr>
</table>
</center></div><p align="center"><font color="#FFFF00"
size="4" face="Bookman Old Style"><strong>D. RISK MANAGEMENT</strong></font><font
color="#FFFF00"><strong><br>
</strong></font></p>
<div align="center"><center><table border="0" width="100%">
  <tr>
    <td width="11%" bgcolor="#808000"
style="border-bottom: medium none"><p align="center"><font
color="#FFFFFF" size="4" face="Times New Roman"><strong>23.</strong></font></p>
    </td>
    <td align="center" width="55%" bgcolor="#808000"
style="border-bottom: medium none"><font
color="#FFFFFF" size="4">Command leaders permit
cutting corners to get a job done.</font></td>
    <td align="center" width="34%" bgcolor="#808000"
style="border-bottom: medium none"><p align="center"><select
name="question23" size="1">
      <option value="X">Select from the following:</option>
      <option value="0">Strongly Disagree</option>
      <option value="1">Moderately Disagree</option>
      <option value="2">Slightly Disagree</option>
      <option value="3">Neutral</option>
      <option value="4">Slightly Agree</option>
      <option value="5">Moderately Agree</option>
      <option value="6">Strongly Agree</option>
      <option value="-1">Not Applicable</option>
    </select></p>
    </td>
  </tr>
</table>
</center></div><hr>
```

```html
<div align="center"><center><table border="0" width="100%">
  <tr>
    <td width="11%" bgcolor="#C0C0C0"
    style="border-bottom: medium none"><p align="center"><font
    color="#000000" size="4" face="Times New Roman"><strong>24.</strong></font></p>
    </td>
    <td align="center" width="55%" bgcolor="#C0C0C0"
    style="border-bottom: medium none"><font
    color="#000000" size="4">Lack of experienced
    personnel has adversely affected my command's ability
    to operate safely.</font></td>
    <td align="center" width="34%" bgcolor="#C0C0C0"
    style="border-bottom: medium none"><p align="center"><select
    name="question24" size="1">
      <option value="X">Select from the following:</option>
      <option value="0">Strongly Disagree</option>
      <option value="1">Moderately Disagree</option>
      <option value="2">Slightly Disagree</option>
      <option value="3">Neutral</option>
      <option value="4">Slightly Agree</option>
      <option value="5">Moderately Agree</option>
      <option value="6">Strongly Agree</option>
      <option value="-1">Not Applicable</option>
    </select></p>
    </td>
  </tr>
</table>
</center></div><hr>
<div align="center"><center><table border="0" width="100%">
  <tr>
    <td width="11%" bgcolor="#808000"
    style="border-bottom: medium none"><p align="center"><font
    color="#FFFFFF" size="4" face="Times New Roman"><strong>25.</strong></font></p>
    </td>
    <td align="center" width="55%" bgcolor="#808000"
    style="border-bottom: medium none"><font
    color="#FFFFFF" size="4">Safety decisions are made at
    the proper levels, by the most qualified people in my
    command.</font></td>
    <td align="center" width="34%" bgcolor="#808000"
    style="border-bottom: medium none"><p align="center"><select
    name="question25" size="1">
      <option value="X">Select from the following:</option>
      <option value="0">Strongly Disagree</option>
      <option value="1">Moderately Disagree</option>
      <option value="2">Slightly Disagree</option>
      <option value="3">Neutral</option>
      <option value="4">Slightly Agree</option>
      <option value="5">Moderately Agree</option>
      <option value="6">Strongly Agree</option>
      <option value="-1">Not Applicable</option>
    </select></p>
    </td>
  </tr>
</table>
</center></div><hr>
```

```html
<div align="center"><center><table border="0" width="100%">
  <tr>
    <td width="11%" bgcolor="#C0C0C0"
    style="border-bottom: medium none"><p align="center"><font
    color="#000000" size="4" face="Times New Roman"><strong>26.</strong></font></p>
    </td>
    <td align="center" width="55%" bgcolor="#C0C0C0"
    style="border-bottom: medium none"><font
    color="#000000" size="4">Command leaders consider
    safety issues during the formation and execution of
    operational and training plans.</font></td>
    <td align="center" width="34%" bgcolor="#C0C0C0"
    style="border-bottom: medium none"><p align="center"><select
    name="question26" size="1">
      <option value="X">Select from the following:</option>
      <option value="0">Strongly Disagree</option>
      <option value="1">Moderately Disagree</option>
      <option value="2">Slightly Disagree</option>
      <option value="3">Neutral</option>
      <option value="4">Slightly Agree</option>
      <option value="5">Moderately Agree</option>
      <option value="6">Strongly Agree</option>
      <option value="-1">Not Applicable</option>
    </select></p>
    </td>
  </tr>
</table>
</center></div><hr>
<div align="center"><center><table border="0" width="100%">
  <tr>
    <td width="11%" bgcolor="#808000"
    style="border-bottom: medium none"><p align="center"><font
    color="#FFFFFF" size="4" face="Times New Roman"><strong>27.</strong></font></p>
    </td>
    <td align="center" width="55%" bgcolor="#808000"
    style="border-bottom: medium none"><font
    color="#FFFFFF" size="4">Command leadership has a
    clear picture of the risks associated with its flight
    operations.</font></td>
    <td align="center" width="34%" bgcolor="#808000"
    style="border-bottom: medium none"><p align="center"><select
    name="question27" size="1">
      <option value="X">Select from the following:</option>
      <option value="0">Strongly Disagree</option>
      <option value="1">Moderately Disagree</option>
      <option value="2">Slightly Disagree</option>
      <option value="3">Neutral</option>
      <option value="4">Slightly Agree</option>
      <option value="5">Moderately Agree</option>
      <option value="6">Strongly Agree</option>
      <option value="-1">Not Applicable</option>
    </select></p>
    </td>
  </tr>
</table>
</center></div><hr>
```

```html
<div align="center"><center><table border="0" width="100%">
  <tr>
    <td width="11%" bgcolor="#C0C0C0"
    style="border-bottom: medium none"><p align="center"><font
    color="#000000" size="4" face="Times New Roman"><strong>28.</strong></font></p>
    </td>
    <td align="center" width="55%" bgcolor="#C0C0C0"
    style="border-bottom: medium none"><font
    color="#000000" size="4">My command takes the time to
    identify and assess risks associated with its flight
    operations.</font></td>
    <td align="center" width="34%" bgcolor="#C0C0C0"
    style="border-bottom: medium none"><p align="center"><select
    name="question28" size="1">
      <option value="X">Select from the following:</option>
      <option value="0">Strongly Disagree</option>
      <option value="1">Moderately Disagree</option>
      <option value="2">Slightly Disagree</option>
      <option value="3">Neutral</option>
      <option value="4">Slightly Agree</option>
      <option value="5">Moderately Agree</option>
      <option value="6">Strongly Agree</option>
      <option value="-1">Not Applicable</option>
    </select></p>
    </td>
  </tr>
</table>
</center></div><hr>
<div align="center"><center><table border="0" width="100%">
  <tr>
    <td width="11%" bgcolor="#808000"
    style="border-bottom: medium none"><p align="center"><font
    color="#FFFFFF" size="4" face="Times New Roman"><strong>29.</strong></font></p>
    </td>
    <td align="center" width="55%" bgcolor="#808000"
    style="border-bottom: medium none"><font
    color="#FFFFFF" size="4">My command does a good job
    managing risks associated with its flight operations.</font></td>
    <td align="center" width="34%" bgcolor="#808000"
    style="border-bottom: medium none"><p align="center"><select
    name="question29" size="1">
      <option value="X">Select from the following:</option>
      <option value="0">Strongly Disagree</option>
      <option value="1">Moderately Disagree</option>
      <option value="2">Slightly Disagree</option>
      <option value="3">Neutral</option>
      <option value="4">Slightly Agree</option>
      <option value="5">Moderately Agree</option>
      <option value="6">Strongly Agree</option>
      <option value="-1">Not Applicable</option>
    </select></p>
    </td>
  </tr>
</table>
</center></div><hr>
<div align="center"><center><table border="0" width="100%">
```

```html
<tr>
    <td width="11%" bgcolor="#C0C0C0"
    style="border-bottom: medium none"><p align="center"><font
    color="#000000" size="4" face="Times New Roman"><strong>30.</strong></font></p>
    </td>
    <td align="center" width="55%" bgcolor="#C0C0C0"
    style="border-bottom: medium none"><font
    color="#000000" size="4">My command has increased the
    chances of a mishap due to inadequate or incorrect
    risk assessment.</font></td>
    <td align="center" width="34%" bgcolor="#C0C0C0"
    style="border-bottom: medium none"><p align="center"><select
    name="question30" size="1">
        <option value="X">Select from the following:</option>
        <option value="0">Strongly Disagree</option>
        <option value="1">Moderately Disagree</option>
        <option value="2">Slightly Disagree</option>
        <option value="3">Neutral</option>
        <option value="4">Slightly Agree</option>
        <option value="5">Moderately Agree</option>
        <option value="6">Strongly Agree</option>
        <option value="-1">Not Applicable</option>
    </select></p>
    </td>
</tr>
</table>
</center></div><hr>
<div align="center"><center><table border="0" width="100%">
    <tr>
        <td width="11%" bgcolor="#808000"
        style="border-bottom: medium none"><p align="center"><font
        color="#FFFFFF" size="4" face="Times New Roman"><strong>31.</strong></font></p>
        </td>
        <td align="center" width="55%" bgcolor="#808000"
        style="border-bottom: medium none"><font
        color="#FFFFFF" size="4">I am provided adequate
        resources (time, staffing, budget, and equipment) to
        accomplish my job.</font></td>
        <td align="center" width="34%" bgcolor="#808000"
        style="border-bottom: medium none"><p align="center"><select
        name="question31" size="1">
            <option value="X">Select from the following:</option>
            <option value="0">Strongly Disagree</option>
            <option value="1">Moderately Disagree</option>
            <option value="2">Slightly Disagree</option>
            <option value="3">Neutral</option>
            <option value="4">Slightly Agree</option>
            <option value="5">Moderately Agree</option>
            <option value="6">Strongly Agree</option>
            <option value="-1">Not Applicable</option>
        </select></p>
        </td>
    </tr>
</table>
</center></div><hr>
<div align="center"><center><table border="0" width="100%">
```

```html
<tr>
    <td width="11%" bgcolor="#C0C0C0"
    style="border-bottom: medium none"><p align="center"><font
    color="#000000" size="4" face="Times New Roman"><strong>32.</strong></font></p>
    </td>
    <td align="center" width="55%" bgcolor="#C0C0C0"
    style="border-bottom: medium none"><font
    color="#000000" size="4">My command provides the
    right number of flight hours per month for me to fly
    safely.</font></td>
    <td align="center" width="34%" bgcolor="#C0C0C0"
    style="border-bottom: medium none"><p align="center"><select
    name="question32" size="1">
        <option value="X">Select from the following:</option>
        <option value="0">Strongly Disagree</option>
        <option value="1">Moderately Disagree</option>
        <option value="2">Slightly Disagree</option>
        <option value="3">Neutral</option>
        <option value="4">Slightly Agree</option>
        <option value="5">Moderately Agree</option>
        <option value="6">Strongly Agree</option>
        <option value="-1">Not Applicable</option>
    </select></p>
    </td>
</tr>
</table>
</center></div><hr>
<div align="center"><center><table border="0" width="100%">
    <tr>
    <td width="11%" bgcolor="#808000"
    style="border-bottom: medium none"><p align="center"><font
    color="#FFFFFF" size="4" face="Times New Roman"><strong>33.</strong></font></p>
    </td>
    <td align="center" width="55%" bgcolor="#808000"
    style="border-bottom: medium none"><font
    color="#FFFFFF" size="4">I have adequate time to
    prepare for and brief my flights.</font></td>
    <td align="center" width="34%" bgcolor="#808000"
    style="border-bottom: medium none"><p align="center"><select
    name="question33" size="1">
        <option value="X">Select from the following:</option>
        <option value="0">Strongly Disagree</option>
        <option value="1">Moderately Disagree</option>
        <option value="2">Slightly Disagree</option>
        <option value="3">Neutral</option>
        <option value="4">Slightly Agree</option>
        <option value="5">Moderately Agree</option>
        <option value="6">Strongly Agree</option>
        <option value="-1">Not Applicable</option>
    </select></p>
    </td>
</tr>
</table>
</center></div><hr>
<div align="center"><center><table border="0" width="100%">
    <tr>
```

```html
<td width="11%" bgcolor="#C0C0C0"
style="border-bottom: medium none"><p align="center"><font
color="#000000" size="4" face="Times New Roman"><strong>34.</strong></font></p>
</td>
<td align="center" width="55%" bgcolor="#C0C0C0"
style="border-bottom: medium none"><font
color="#000000" size="4">Based upon my command's
personnel and other assets, the command is
over-committed.</font></td>
<td align="center" width="34%" bgcolor="#C0C0C0"
style="border-bottom: medium none"><p align="center"><select
name="question34" size="1">
    <option value="X">Select from the following:</option>
    <option value="0">Strongly Disagree</option>
    <option value="1">Moderately Disagree</option>
    <option value="2">Slightly Disagree</option>
    <option value="3">Neutral</option>
    <option value="4">Slightly Agree</option>
    <option value="5">Moderately Agree</option>
    <option value="6">Strongly Agree</option>
    <option value="-1">Not Applicable</option>
  </select></p>
  </td>
 </tr>
</table>
</center></div><hr>
<div align="center"><center><table border="0" width="100%">
  <tr>
    <td width="11%" bgcolor="#808000"
style="border-bottom: medium none"><p align="center"><font
color="#FFFFFF" size="4" face="Times New Roman"><strong>35.</strong></font></p>
    </td>
    <td align="center" width="55%" bgcolor="#808000"
style="border-bottom: medium none"><font
color="#FFFFFF" size="4">My command has incorporated
Operational Risk Management processes in
decision-making at all levels.</font></td>
    <td align="center" width="34%" bgcolor="#808000"
style="border-bottom: medium none"><p align="center"><select
name="question35" size="1">
    <option value="X">Select from the following:</option>
    <option value="0">Strongly Disagree</option>
    <option value="1">Moderately Disagree</option>
    <option value="2">Slightly Disagree</option>
    <option value="3">Neutral</option>
    <option value="4">Slightly Agree</option>
    <option value="5">Moderately Agree</option>
    <option value="6">Strongly Agree</option>
    <option value="-1">Not Applicable</option>
  </select></p>
  </td>
 </tr>
</table>
</center></div><hr>
<div align="center"><center><table border="0" width="100%">
  <tr>
```

```html
        <td width="11%" bgcolor="#C0C0C0"
        style="border-bottom: medium none"><p align="center"><font
        color="#000000" size="4" face="Times New Roman"><strong>36.</strong></font></p>
        </td>
        <td align="center" width="55%" bgcolor="#C0C0C0"
        style="border-bottom: medium none"><font
        color="#000000" size="4">My supervisor can be relied
        on to keep his/her word.</font></td>
        <td align="center" width="34%" bgcolor="#C0C0C0"
        style="border-bottom: medium none"><p align="center"><select
        name="question36" size="1">
            <option value="X">Select from the following:</option>
            <option value="0">Strongly Disagree</option>
            <option value="1">Moderately Disagree</option>
            <option value="2">Slightly Disagree</option>
            <option value="3">Neutral</option>
            <option value="4">Slightly Agree</option>
            <option value="5">Moderately Agree</option>
            <option value="6">Strongly Agree</option>
            <option value="-1">Not Applicable</option>
        </select></p>
        </td>
    </tr>
</table>
</center></div><hr>
<div align="center"><center><table border="0" width="100%">
    <tr>
        <td width="11%" bgcolor="#808000"
        style="border-bottom: medium none"><p align="center"><font
        color="#FFFFFF" size="4" face="Times New Roman"><strong>37.</strong></font></p>
        </td>
        <td align="center" width="55%" bgcolor="#808000"
        style="border-bottom: medium none"><font
        color="#FFFFFF" size="4">Our command leaders and
        supervisors can be trusted.</font></td>
        <td align="center" width="34%" bgcolor="#808000"
        style="border-bottom: medium none"><p align="center"><select
        name="question37" size="1">
            <option value="X">Select from the following:</option>
            <option value="0">Strongly Disagree</option>
            <option value="1">Moderately Disagree</option>
            <option value="2">Slightly Disagree</option>
            <option value="3">Neutral</option>
            <option value="4">Slightly Agree</option>
            <option value="5">Moderately Agree</option>
            <option value="6">Strongly Agree</option>
            <option value="-1">Not Applicable</option>
        </select></p>
        </td>
    </tr>
</table>
</center></div><hr>
<div align="center"><center><table border="0" width="100%">
    <tr>
        <td width="11%" bgcolor="#C0C0C0"
        style="border-bottom: medium none"><p align="center"><font
```

```html
color="#000000" size="4" face="Times New Roman"><strong>38.</strong></font></p>
</td>
<td align="center" width="55%" bgcolor="#C0C0C0"
style="border-bottom: medium none"><font
color="#000000" size="4">My command's Safety Officer
is highly regarded.</font></td>
<td align="center" width="34%" bgcolor="#C0C0C0"
style="border-bottom: medium none"><p align="center"><select
name="question38" size="1">
    <option value="X">Select from the following:</option>
    <option value="0">Strongly Disagree</option>
    <option value="1">Moderately Disagree</option>
    <option value="2">Slightly Disagree</option>
    <option value="3">Neutral</option>
    <option value="4">Slightly Agree</option>
    <option value="5">Moderately Agree</option>
    <option value="6">Strongly Agree</option>
    <option value="-1">Not Applicable</option>
</select></p>
</td>
  </tr>
</table>
</center></div><hr>
<div align="center"><center><table border="0" width="100%">
  <tr>
    <td width="11%" bgcolor="#808000"
style="border-bottom: medium none"><p align="center"><font
color="#FFFFFF" size="4" face="Times New Roman"><strong>39.</strong></font></p>
</td>
    <td align="center" width="55%" bgcolor="#808000"
style="border-bottom: medium none"><font
color="#FFFFFF" size="4">Our Safety Officer is
influential in promoting safety.</font></td>
    <td align="center" width="34%" bgcolor="#808000"
style="border-bottom: medium none"><p align="center"><select
name="question39" size="1">
    <option value="X">Select from the following:</option>
    <option value="0">Strongly Disagree</option>
    <option value="1">Moderately Disagree</option>
    <option value="2">Slightly Disagree</option>
    <option value="3">Neutral</option>
    <option value="4">Slightly Agree</option>
    <option value="5">Moderately Agree</option>
    <option value="6">Strongly Agree</option>
    <option value="-1">Not Applicable</option>
</select></p>
</td>
  </tr>
</table>
</center></div><hr>
<div align="center"><center><table border="0" width="100%">
  <tr>
    <td width="11%" bgcolor="#C0C0C0"
style="border-bottom: medium none"><p align="center"><font
color="#000000" size="4" face="Times New Roman"><strong>40.</strong></font></p>
</td>
```

```html
<td align="center" width="55%" bgcolor="#C0C0C0"
style="border-bottom: medium none"><font
color="#000000" size="4">My command is genuinely
concerned about safety.</font></td>
<td align="center" width="34%" bgcolor="#C0C0C0"
style="border-bottom: medium none"><p align="center"><select
name="question40" size="1">
    <option value="X">Select from the following:</option>
    <option value="0">Strongly Disagree</option>
    <option value="1">Moderately Disagree</option>
    <option value="2">Slightly Disagree</option>
    <option value="3">Neutral</option>
    <option value="4">Slightly Agree</option>
    <option value="5">Moderately Agree</option>
    <option value="6">Strongly Agree</option>
    <option value="-1">Not Applicable</option>
  </select></p>
  </td>
</tr>
</table>
</center></div><hr>
<div align="center"><center><table border="0" width="100%">
  <tr>
    <td width="11%" bgcolor="#808000"
style="border-bottom: medium none"><p align="center"><font
color="#FFFFFF" size="4" face="Times New Roman"><strong>41.</strong></font></p>
    </td>
    <td align="center" width="55%" bgcolor="#808000"
style="border-bottom: medium none"><font
color="#FFFFFF" size="4">Command leadership is
successful in communicating its safety goals to unit
personnel.</font></td>
    <td align="center" width="34%" bgcolor="#808000"
style="border-bottom: medium none"><p align="center"><select
name="question41" size="1">
    <option value="X">Select from the following:</option>
    <option value="0">Strongly Disagree</option>
    <option value="1">Moderately Disagree</option>
    <option value="2">Slightly Disagree</option>
    <option value="3">Neutral</option>
    <option value="4">Slightly Agree</option>
    <option value="5">Moderately Agree</option>
    <option value="6">Strongly Agree</option>
    <option value="-1">Not Applicable</option>
  </select></p>
  </td>
</tr>
</table>
</center></div><hr>
<div align="center"><center><table border="0" width="100%">
  <tr>
    <td width="11%" bgcolor="#C0C0C0"
style="border-bottom: medium none"><p align="center"><font
color="#000000" size="4" face="Times New Roman"><strong>42.</strong></font></p>
    </td>
    <td align="center" width="55%" bgcolor="#C0C0C0"
```

```html
style="border-bottom: medium none"><font
color="#000000" size="4">My command provides a
positive command climate that promotes safe flight
operations.</font></td>
<td align="center" width="34%" bgcolor="#C0C0C0"
style="border-bottom: medium none"><p align="center"><select
name="question42" size="1">
    <option value="X">Select from the following:</option>
    <option value="0">Strongly Disagree</option>
    <option value="1">Moderately Disagree</option>
    <option value="2">Slightly Disagree</option>
    <option value="3">Neutral</option>
    <option value="4">Slightly Agree</option>
    <option value="5">Moderately Agree</option>
    <option value="6">Strongly Agree</option>
    <option value="-1">Not Applicable</option>
</select></p>
</td>
</tr>
</table>
</center></div><hr>
<div align="center"><center><table border="0" width="100%">
<tr>
    <td width="11%" bgcolor="#808000"
style="border-bottom: medium none"><p align="center"><font
color="#FFFFFF" size="4" face="Times New Roman"><strong>43.</strong></font></p>
</td>
    <td align="center" width="55%" bgcolor="#808000"
style="border-bottom: medium none"><font
color="#FFFFFF" size="4">Command leadership is
actively involved in the safety program and
management of safety matters.</font></td>
    <td align="center" width="34%" bgcolor="#808000"
style="border-bottom: medium none"><p align="center"><select
name="question43" size="1">
    <option value="X">Select from the following:</option>
    <option value="0">Strongly Disagree</option>
    <option value="1">Moderately Disagree</option>
    <option value="2">Slightly Disagree</option>
    <option value="3">Neutral</option>
    <option value="4">Slightly Agree</option>
    <option value="5">Moderately Agree</option>
    <option value="6">Strongly Agree</option>
    <option value="-1">Not Applicable</option>
</select></p>
</td>
</tr>
</table>
</center></div><hr>
<div align="center"><center><table border="0" width="100%">
<tr>
    <td width="11%" bgcolor="#C0C0C0"
style="border-bottom: medium none"><p align="center"><font
color="#000000" size="4" face="Times New Roman"><strong>44.</strong></font></p>
</td>
    <td align="center" width="55%" bgcolor="#C0C0C0"
```

```html
style="border-bottom: medium none"><font
color="#000000" size="4">Command leadership sets the
example for compliance with flight standards.</font></td>
<td align="center" width="34%" bgcolor="#C0C0C0"
style="border-bottom: medium none"><p align="center"><select
name="question44" size="1">
    <option value="X">Select from the following:</option>
    <option value="0">Strongly Disagree</option>
    <option value="1">Moderately Disagree</option>
    <option value="2">Slightly Disagree</option>
    <option value="3">Neutral</option>
    <option value="4">Slightly Agree</option>
    <option value="5">Moderately Agree</option>
    <option value="6">Strongly Agree</option>
    <option value="-1">Not Applicable</option>
  </select></p>
  </td>
  </tr>
</table>
</center></div><hr>
<div align="center"><center><table border="0" width="100%">
  <tr>
    <td width="11%" bgcolor="#808000"
style="border-bottom: medium none"><p align="center"><font
color="#FFFFFF" size="4" face="Times New Roman"><strong>45.</strong></font></p>
    </td>
    <td align="center" width="55%" bgcolor="#808000"
style="border-bottom: medium none"><font
color="#FFFFFF" size="4">My command ensures that all
unit members are responsible and accountable for safe
flight operations.</font></td>
    <td align="center" width="34%" bgcolor="#808000"
style="border-bottom: medium none"><p align="center"><select
name="question45" size="1">
    <option value="X">Select from the following:</option>
    <option value="0">Strongly Disagree</option>
    <option value="1">Moderately Disagree</option>
    <option value="2">Slightly Disagree</option>
    <option value="3">Neutral</option>
    <option value="4">Slightly Agree</option>
    <option value="5">Moderately Agree</option>
    <option value="6">Strongly Agree</option>
    <option value="-1">Not Applicable</option>
  </select></p>
  </td>
  </tr>
</table>
</center></div><hr>
<div align="center"><center><table border="0" width="100%">
  <tr>
    <td width="11%" bgcolor="#C0C0C0"
style="border-bottom: medium none"><p align="center"><font
color="#000000" size="4" face="Times New Roman"><strong>46.</strong></font></p>
    </td>
    <td align="center" width="55%" bgcolor="#C0C0C0"
style="border-bottom: medium none"><font
```

```
color="#000000" size="4">Command leadership willingly
assists in providing advice concerning safety
matters.</font></td>
<td align="center" width="34%" bgcolor="#C0C0C0"
style="border-bottom: medium none"><p align="center"><select
name="question46" size="1">
    <option value="X">Select from the following:</option>
    <option value="0">Strongly Disagree</option>
    <option value="1">Moderately Disagree</option>
    <option value="2">Slightly Disagree</option>
    <option value="3">Neutral</option>
    <option value="4">Slightly Agree</option>
    <option value="5">Moderately Agree</option>
    <option value="6">Strongly Agree</option>
    <option value="-1">Not Applicable</option>
</select></p>
</td>
</tr>
</table>
</center></div><hr>
<div align="center"><center><table border="0" width="100%">
<tr>
<td width="11%" bgcolor="#808000"
style="border-bottom: medium none"><p align="center"><font
color="#FFFFFF" size="4" face="Times New Roman"><strong>47.</strong></font></p>
</td>
<td align="center" width="55%" bgcolor="#808000"
style="border-bottom: medium none"><font
color="#FFFFFF" size="4">Command leadership reacts
well to unexpected changes to its plans.</font></td>
<td align="center" width="34%" bgcolor="#808000"
style="border-bottom: medium none"><p align="center"><select
name="question47" size="1">
    <option value="X">Select from the following:</option>
    <option value="0">Strongly Disagree</option>
    <option value="1">Moderately Disagree</option>
    <option value="2">Slightly Disagree</option>
    <option value="3">Neutral</option>
    <option value="4">Slightly Agree</option>
    <option value="5">Moderately Agree</option>
    <option value="6">Strongly Agree</option>
    <option value="-1">Not Applicable</option>
</select></p>
</td>
</tr>
</table>
</center></div><hr>
<div align="center"><center><table border="0" width="100%">
<tr>
<td width="11%" bgcolor="#C0C0C0"
style="border-bottom: medium none"><p align="center"><font
color="#000000" size="4" face="Times New Roman"><strong>48.</strong></font></p>
</td>
<td align="center" width="55%" bgcolor="#C0C0C0"
style="border-bottom: medium none"><font
color="#000000" size="4">My command does not hesitate
```

```html
to temporarily restrict from flying individuals who
are under high personal stress.</font></td>
<td align="center" width="34%" bgcolor="#C0C0C0"
style="border-bottom: medium none"><p align="center"><select
name="question48" size="1">
    <option value="X">Select from the following:</option>
    <option value="0">Strongly Disagree</option>
    <option value="1">Moderately Disagree</option>
    <option value="2">Slightly Disagree</option>
    <option value="3">Neutral</option>
    <option value="4">Slightly Agree</option>
    <option value="5">Moderately Agree</option>
    <option value="6">Strongly Agree</option>
    <option value="-1">Not Applicable</option>
</select></p>
</td>
  </tr>
</table>
</center></div><hr>
<div align="center"><center><table border="0" width="100%">
  <tr>
    <td width="11%" bgcolor="#808000"
style="border-bottom: medium none"><p align="center"><font
color="#FFFFFF" size="4" face="Times New Roman"><strong>49.</strong></font></p>
    </td>
    <td align="center" width="55%" bgcolor="#808000"
style="border-bottom: medium none"><font
color="#FFFFFF" size="4">I am adequately trained to
safely conduct all of my flights.</font></td>
    <td align="center" width="34%" bgcolor="#808000"
style="border-bottom: medium none"><p align="center"><select
name="question49" size="1">
    <option value="X">Select from the following:</option>
    <option value="0">Strongly Disagree</option>
    <option value="1">Moderately Disagree</option>
    <option value="2">Slightly Disagree</option>
    <option value="3">Neutral</option>
    <option value="4">Slightly Agree</option>
    <option value="5">Moderately Agree</option>
    <option value="6">Strongly Agree</option>
    <option value="-1">Not Applicable</option>
</select></p>
    </td>
  </tr>
</table>
</center></div><hr>
<div align="center"><center><table border="0" width="100%">
  <tr>
    <td width="11%" bgcolor="#C0C0C0"
style="border-bottom: medium none"><p align="center"><font
color="#FFFFFF" size="4" face="Times New Roman"><strong>50.</strong></font></p>
    </td>
    <td align="center" width="55%" bgcolor="#C0C0C0"
style="border-bottom: medium none"><font
color="#000000" size="4">Morale and motivation in my
command are high.</font></td>
```

```html
<td align="center" width="34%" bgcolor="#C0C0C0"
style="border-bottom: medium none"><p align="center"><select
name="question50" size="1">
    <option value="X">Select from the following:</option>
    <option value="0">Strongly Disagree</option>
    <option value="1">Moderately Disagree</option>
    <option value="2">Slightly Disagree</option>
    <option value="3">Neutral</option>
    <option value="4">Slightly Agree</option>
    <option value="5">Moderately Agree</option>
    <option value="6">Strongly Agree</option>
    <option value="-1">Not Applicable</option>
  </select></p>
  </td>
 </tr>
</table>
</center></div><hr>
<div align="center"><center><table border="0" width="100%">
  <tr>
    <td width="11%" bgcolor="#808000"
style="border-bottom: medium none"><p align="center"><font
color="#FFFFFF" size="4" face="Times New Roman"><strong>51.</strong></font></p>
    </td>
    <td align="center" width="55%" bgcolor="#808000"
style="border-bottom: medium none"><font
color="#FFFFFF" size="4">My command ensures the
uniform enforcement of all operating standards among
unit members.</font></td>
    <td align="center" width="34%" bgcolor="#808000"
style="border-bottom: medium none"><p align="center"><select
name="question51" size="1">
    <option value="X">Select from the following:</option>
    <option value="0">Strongly Disagree</option>
    <option value="1">Moderately Disagree</option>
    <option value="2">Slightly Disagree</option>
    <option value="3">Neutral</option>
    <option value="4">Slightly Agree</option>
    <option value="5">Moderately Agree</option>
    <option value="6">Strongly Agree</option>
    <option value="-1">Not Applicable</option>
  </select></p>
  </td>
 </tr>
</table>
</center></div><hr>
<div align="center"><center><table border="0" width="100%">
  <tr>
    <td width="11%" bgcolor="#C0C0C0"
style="border-bottom: medium none"><p align="center"><font
color="#000000" size="4" face="Times New Roman"><strong>52.</strong></font></p>
    </td>
    <td align="center" width="55%" bgcolor="#C0C0C0"
style="border-bottom: medium none"><font
color="#000000" size="4">Crew rest standards are
enforced in my command.</font></td>
    <td align="center" width="34%" bgcolor="#C0C0C0"
```

```
                style="border-bottom: medium none"><p align="center"><select
                name="question52" size="1">
                    <option value="X">Select from the following:</option>
                    <option value="0">Strongly Disagree</option>
                    <option value="1">Moderately Disagree</option>
                    <option value="2">Slightly Disagree</option>
                    <option value="3">Neutral</option>
                    <option value="4">Slightly Agree</option>
                    <option value="5">Moderately Agree</option>
                    <option value="6">Strongly Agree</option>
                    <option value="-1">Not Applicable</option>
                </select></p>
                </td>
          </tr>
      </table>
      </center></div><hr>
      <div align="center"><center><table border="0" width="100%">
          <tr>
              <td width="11%" bgcolor="#808000"
              style="border-bottom: medium none"><p align="center"><font
              color="#FFFFFF" size="4" face="Times New Roman"><strong>53.</strong></font></p>
              </td>
              <td align="center" width="55%" bgcolor="#808000"
              style="border-bottom: medium none"><font
              color="#FFFFFF" size="4">In my command, NATOPS tests
              and check rides are conducted as intended, to
              candidly assess aircrew qualifications.</font></td>
              <td align="center" width="34%" bgcolor="#808000"
              style="border-bottom: medium none"><p align="center"><select
              name="question53" size="1">
                  <option value="X">Select from the following:</option>
                  <option value="0">Strongly Disagree</option>
                  <option value="1">Moderately Disagree</option>
                  <option value="2">Slightly Disagree</option>
                  <option value="3">Neutral</option>
                  <option value="4">Slightly Agree</option>
                  <option value="5">Moderately Agree</option>
                  <option value="6">Strongly Agree</option>
                  <option value="-1">Not Applicable</option>
              </select></p>
              </td>
          </tr>
      </table>
      </center></div><hr>
      <div align="center"><center><table border="0" width="100%">
          <tr>
              <td width="11%" bgcolor="#C0C0C0"
              style="border-bottom: medium none"><p align="center"><font
              color="#000000" size="4" face="Times New Roman"><strong>54.</strong></font></p>
              </td>
              <td align="center" width="55%" bgcolor="#C0C0C0"
              style="border-bottom: medium none"><font
              color="#000000" size="4">My command provides adequate
              safety backups to catch possible human errors during
              high-risk missions.</font></td>
              <td align="center" width="34%" bgcolor="#C0C0C0"
```

```
style="border-bottom: medium none"><p align="center"><select
name="question54" size="1">
    <option value="X">Select from the following:</option>
    <option value="0">Strongly Disagree</option>
    <option value="1">Moderately Disagree</option>
    <option value="2">Slightly Disagree</option>
    <option value="3">Neutral</option>
    <option value="4">Slightly Agree</option>
    <option value="5">Moderately Agree</option>
    <option value="6">Strongly Agree</option>
    <option value="-1">Not Applicable</option>
  </select></p>
  </td>
  </tr>
</table>
</center></div><hr>
<div align="center"><center><table border="0" width="100%">
  <tr>
    <td width="11%" bgcolor="#808000"
    style="border-bottom: medium none"><p align="center"><font
    color="#FFFFFF" size="4" face="Times New Roman"><strong>55.</strong></font></p>
    </td>
    <td align="center" width="55%" bgcolor="#808000"
    style="border-bottom: medium none"><font
    color="#FFFFFF" size="4">Within my command, good
    communications flow exists up and down the chain of
    command.</font></td>
    <td align="center" width="34%" bgcolor="#808000"
    style="border-bottom: medium none"><p align="center"><select
    name="question55" size="1">
      <option value="X">Select from the following:</option>
      <option value="0">Strongly Disagree</option>
      <option value="1">Moderately Disagree</option>
      <option value="2">Slightly Disagree</option>
      <option value="3">Neutral</option>
      <option value="4">Slightly Agree</option>
      <option value="5">Moderately Agree</option>
      <option value="6">Strongly Agree</option>
      <option value="-1">Not Applicable</option>
    </select></p>
    </td>
  </tr>
</table>
</center></div><hr>
<div align="center"><center><table border="0" width="100%">
  <tr>
    <td width="11%" bgcolor="#C0C0C0"
    style="border-bottom: medium none"><p align="center"><font
    color="#000000" size="4" face="Times New Roman"><strong>56.</strong></font></p>
    </td>
    <td align="center" width="55%" bgcolor="#C0C0C0"
    style="border-bottom: medium none"><font
    color="#000000" size="4">My command has good two-way
    communication with external commands.</font></td>
    <td align="center" width="34%" bgcolor="#C0C0C0"
    style="border-bottom: medium none"><p align="center"><select
```

```html
        name="question56" size="1">
            <option value="X">Select from the following:</option>
            <option value="0">Strongly Disagree</option>
            <option value="1">Moderately Disagree</option>
            <option value="2">Slightly Disagree</option>
            <option value="3">Neutral</option>
            <option value="4">Slightly Agree</option>
            <option value="5">Moderately Agree</option>
            <option value="6">Strongly Agree</option>
            <option value="-1">Not Applicable</option>
        </select></p>
        </td>
    </tr>
</table>
</center></div><hr>
<div align="center"><center><table border="0" width="100%">
    <tr>
        <td width="11%" bgcolor="#808000"
        style="border-bottom: medium none"><p align="center"><font
        color="#FFFFFF" size="4" face="Times New Roman"><strong>57.</strong></font></p>
        </td>
        <td align="center" width="55%" bgcolor="#808000"
        style="border-bottom: medium none"><font
        color="#FFFFFF" size="4">Safety education and
        training are adequate in my command.</font></td>
        <td align="center" width="34%" bgcolor="#808000"
        style="border-bottom: medium none"><p align="center"><select
        name="question57" size="1">
            <option value="X">Select from the following:</option>
            <option value="0">Strongly Disagree</option>
            <option value="1">Moderately Disagree</option>
            <option value="2">Slightly Disagree</option>
            <option value="3">Neutral</option>
            <option value="4">Slightly Agree</option>
            <option value="5">Moderately Agree</option>
            <option value="6">Strongly Agree</option>
            <option value="-1">Not Applicable</option>
        </select></p>
        </td>
    </tr>
</table>
</center></div><hr>
<div align="center"><center><table border="0" width="100%">
    <tr>
        <td width="11%" bgcolor="#C0C0C0"
        style="border-bottom: medium none"><p align="center"><font
        color="#000000" size="4" face="Times New Roman"><strong>58.</strong></font></p>
        </td>
        <td align="center" width="55%" bgcolor="#C0C0C0"
        style="border-bottom: medium none"><font
        color="#000000" size="4">The Safety Department is a
        well-respected element of my command.</font></td>
        <td align="center" width="34%" bgcolor="#C0C0C0"
        style="border-bottom: medium none"><p align="center"><select
        name="question58" size="1">
            <option value="X">Select from the following:</option>
```

```html
      <option value="0">Strongly Disagree</option>
      <option value="1">Moderately Disagree</option>
      <option value="2">Slightly Disagree</option>
      <option value="3">Neutral</option>
      <option value="4">Slightly Agree</option>
      <option value="5">Moderately Agree</option>
      <option value="6">Strongly Agree</option>
      <option value="-1">Not Applicable</option>
    </select></p>
    </td>
  </tr>
</table>
</center></div><hr>
<div align="center"><center><table border="0" width="100%">
  <tr>
    <td width="11%" bgcolor="#808000"
    style="border-bottom: medium none"><p align="center"><font
    color="#FFFFFF" size="4" face="Times New Roman"><strong>59.</strong></font></p>
    </td>
    <td align="center" width="55%" bgcolor="#808000"
    style="border-bottom: medium none"><font
    color="#FFFFFF" size="4">The Aviation Safety Officer
    position is a sought after billet in my command.</font></td>
    <td align="center" width="34%" bgcolor="#808000"
    style="border-bottom: medium none"><p align="center"><select
    name="question59" size="1">
      <option value="X">Select from the following:</option>
      <option value="0">Strongly Disagree</option>
      <option value="1">Moderately Disagree</option>
      <option value="2">Slightly Disagree</option>
      <option value="3">Neutral</option>
      <option value="4">Slightly Agree</option>
      <option value="5">Moderately Agree</option>
      <option value="6">Strongly Agree</option>
      <option value="-1">Not Applicable</option>
    </select></p>
    </td>
  </tr>
</table>
</center></div><hr>
<div align="center"><center><table border="0" width="100%">
  <tr>
    <td width="11%" bgcolor="#C0C0C0"
    style="border-bottom: medium none"><p align="center"><font
    color="#000000" size="4" face="Times New Roman"><strong>60.</strong></font></p>
    </td>
    <td align="center" width="55%" bgcolor="#C0C0C0"
    style="border-bottom: medium none"><font
    color="#000000" size="4">My command's Safety
    Department keeps me well informed regarding important
    safety information.</font></td>
    <td align="center" width="34%" bgcolor="#C0C0C0"
    style="border-bottom: medium none"><p align="center"><select
    name="question60" size="1">
      <option value="X">Select from the following:</option>
      <option value="0">Strongly Disagree</option>
```

```html
        <option value="1">Moderately Disagree</option>
        <option value="2">Slightly Disagree</option>
        <option value="3">Neutral</option>
        <option value="4">Slightly Agree</option>
        <option value="5">Moderately Agree</option>
        <option value="6">Strongly Agree</option>
        <option value="-1">Not Applicable</option>
      </select></p>
      </td>
    </tr>
  </table>
  </center></div><hr>
  <div align="center"><center><table border="0" width="100%">
    <tr>
      <td width="11%" bgcolor="#808000"
      style="border-bottom: medium none"><p align="center"><font
      color="#FFFFFF" size="4" face="Times New Roman"><strong>61.</strong></font></p>
      </td>
      <td align="center" width="55%" bgcolor="#808000"
      style="border-bottom: medium none"><font
      color="#000000" size="4">My command's Aircrew
      Coordination Training program is helping to improve
      mission performance and safety.</font></td>
      <td align="center" width="34%" bgcolor="#808000"
      style="border-bottom: medium none"><p align="center"><select
      name="question61" size="1">
        <option value="X">Select from the following:</option>
        <option value="0">Strongly Disagree</option>
        <option value="1">Moderately Disagree</option>
        <option value="2">Slightly Disagree</option>
        <option value="3">Neutral</option>
        <option value="4">Slightly Agree</option>
        <option value="5">Moderately Agree</option>
        <option value="6">Strongly Agree</option>
        <option value="-1">Not Applicable</option>
      </select></p>
      </td>
    </tr>
  </table>
  </center></div><p align="center"> </p>
  <div align="center"><table border="0" width="100%">
    <tr>
      <td width="50%"><center><input
      type="submit" name="submit" value="Submit">
            </p></center>
      </td>
      <td width="50%"><center>     
      <input type="reset" name="reset" value=" Reset "> </td></center>
    </tr>
  </table>
  </div><div align="center"><center><table border="0"
  width="100%">
    <tr>
      <td><p align="center"><!--webbot bot="ImageMap"
      startspan
      rectangle=" (1,5) (52, 38) http://spitfire.avsafety.nps.navy.mil/Help/Help.html##_blank"
```

```
src="images/help.gif" border="0" width="53"
height="39" --><MAP NAME="FrontPageMap1"><AREA SHAPE="RECT" COORDS="1, 5, 52,
38" HREF="http://spitfire.avsafety.nps.navy.mil/Help/Help.html"
TARGET="_blank"></MAP><img src="images/help.gif" border="0" width="53" height="39"
usemap="#FrontPageMap1"><!--webbot bot="ImageMap"
i-checksum="53524" endspan --></p>
    </td>
  </tr>
  </table>
  </center></div>
</form>
<script language="javascript" src="javascript/clearsurvey.js">
</script>
</body>
</html>
```

APPENDIX E. JAVASCRIPT ERROR CHECKING CODE

A. WHY USE JAVASCRIPT?

The HTML form, as presented in Appendix E, is one means by which the client can fill out and submit his survey to the *Spitfire* server for processing. Error checking could be performed once the server receives the information, however, doing so uses invaluable server resources. A heavy load of simultaneous client requests would delay the submission process, and clients would see considerable delay as to whether their input was successfully accepted or not. To avoid this, error checking is done on the client's machine through the use of Javascript.

The HTML form includes a Javascript file - *checksurvey.js* - which performs the necessary checks. When the client clicks the "Submit" button, the Javascript function *check_form* is executed. This method returns a boolean; if true, the form is sent on to the action URL

http://spitfire.avsafety.nps.navy.mil:8080/servlet/SurveyServlet,
otherwise a dialog box appears and informs the client of problems with the submission:
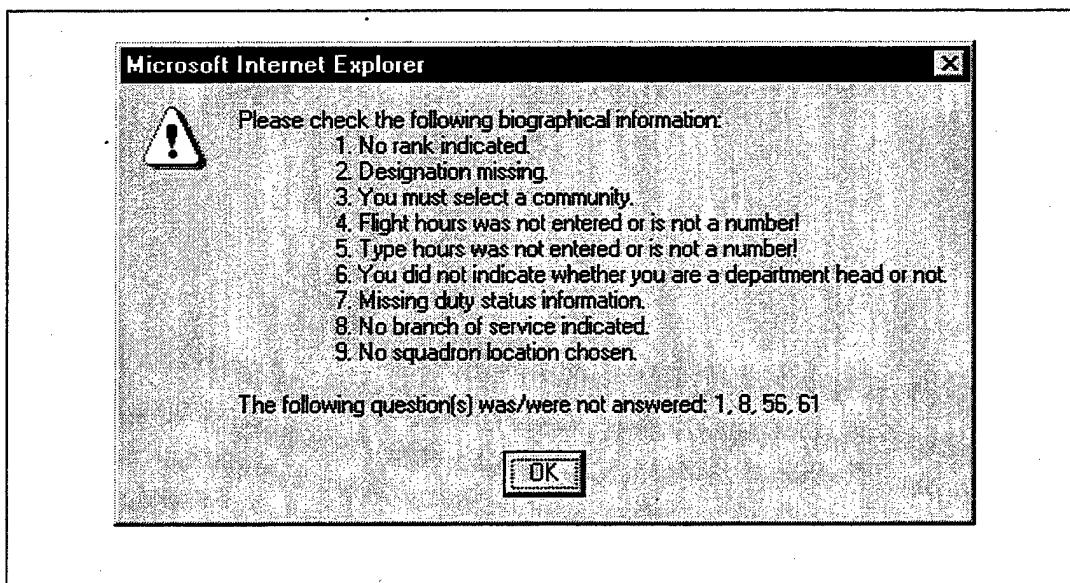
Figure E.1. An Alert Dialog Box - the client is notified of submission errors.

Once the client clicks the "OK" button, he will have the opportunity to correct the survey and submit it again. If there are no problems with the information entered, the client is immediately prompted by two dialog boxes which request the client's squadron and survey numbers. As in the Java™ Survey

applet, this information is used to authenticate the client submitting the form. *SurveyServlet* takes this information, checks the information by performing database queries, and either continues with processing the input or redirects the client to a different page based on the type of error that occurred.

The Javascript code that performs the error checking is included in this Appendix for the reader's reference:

B.   JAVASCRIPT SOURCE CODE

```
//constants used by checksurvey script
var MAX_QUESTIONS = 61;
var FIRST_QUESTION_ELEMENT = 11;

//global boolean
var form_errors = false;

//global count of errors
var error_count = 0;

//initially no errors
var error_messages = new Array();

//-------------------------------------------------------------------------------------
//Function:    check_bio_information()
//Parameters:  None
//Return Value: None
//Purpose:     Checks all of the biographical information the client entered to
//             ensure that values are entered.  Error results are concatenated
//             together to form one error message that is entered into a global array
//             variable for later use.
//-------------------------------------------------------------------------------------
function check_bio_information(){
   var message = "Please check the following biographical information:\n";
   var count = 0;

   //boolean indicating whether there is a problem with the biographical information
   var problem, validHour1, validHour2;

   //if a problem exists with form data, then it was from a previous check, and we need to
   //reset it before checking again
   if (form_errors)
      form_errors = false;

   //check each of the biographical entries for errors
   if (document.survey.rank.selectedIndex == 0){
      problem = true;
      message += "\t" + eval(++count) + ". No rank indicated.\n";
   }
   if (document.survey.designation.selectedIndex == 0){
      problem = true;
      message += "\t" + eval(++count) + ". Designation missing.\n";
   }
   if (document.survey.community.selectedIndex == 0){
```

```javascript
      problem = true;
      message += "\t" + eval(++count) + ". You must select a community.\n";
    }

    validHour1 = check_string(new String(document.survey.flight_hours.value));

    if (!validHour1){
      problem = true;
      message += "\t" + eval(++count) + ". Flight hours was not entered or is not a number!\n";
    }

    validHour2 = check_string(new String(document.survey.type_hours.value));

    if (!validHour2){
      problem = true;
      message += "\t" + eval(++count) + ". Type hours was not entered or is not a number!\n";
    }

    if (validHour1 && validHour2){
      var val1 = parseInt(document.survey.flight_hours.value);
      var val2 = parseInt(document.survey.type_hours.value);
      if (val1 > 25000){
        problem = true;
        message += "\t" + eval(++count) + ". Flight hours must be less than 25000.\n";
      }
      if (val2 > 25000){
        problem = true;
        message += "\t" + eval(++count) + ". Type hours must be less than 25000.\n";
      }
      if (val2 > val1){
        problem = true;
        message += "\t" + eval(++count) + ". Number of type hours cannot be greater than flight hours.\n";
      }
    }

    if (document.survey.department_head.selectedIndex == 0){
      problem = true;
      message += "\t" + eval(++count) + ". You did not indicate whether you are a department head or not.\n";
    }
    if (document.survey.status.selectedIndex == 0){
      problem = true;
      message += "\t" + eval(++count) + ". Missing duty status information.\n";
    }
    if (document.survey.service.selectedIndex == 0){
      problem = true;
      message += "\t" + eval(++count) + ". No branch of service indicated.\n";
    }
    if (document.survey.location.selectedIndex == 0){
      problem = true;
      message += "\t" + eval(++count) + ". No squadron location chosen.";
    }
```

```javascript
    //if there was a problem, enter the message into the error_messages array and set form_errors to true
    if (problem){
        error_messages[error_count++] = message;
        form_errors = true;
    }
}//end check_bio_information()


//-------------------------------------------------------------------------------------------
//Function:    check_string()
//Parameters:  None
//Return Value: boolean, indicating whether text entries are valid numbers or not
//Purpose:     Checks the number of flight and type hours to determine if valid
//             they are valid numbers or not. This is a utility function called by
//             the method check_bio_information().
//-------------------------------------------------------------------------------------------
function check_string(value){

    if (value.length == 0)
        return false;

    var goodChars = new String("0123456789");
    var validChar = true;
    var validEntry = true;

    for (var jx=0; jx<value.length; jx++){
        for (var kx=0; kx<goodChars.length; kx++){
            if (value.charAt(jx) == goodChars.charAt(kx))
                validEntry = validChar = true;
        }
        if (!validChar) ·
            return false;
        validChar = false;
    }

    return validEntry;
}//end check_string()


//-------------------------------------------------------------------------------------------
//Function:    check_survey_questions()
//Parameters:  None
//Return Value: None
//Purpose:     Checks all of the survey questions to make sure they are answered. If
//             a question remains unanswered, the user will be notified when they try
//             to submit the form.
//-------------------------------------------------------------------------------------------
function check_survey_questions(){

    var problem = false;
    var errors = new Array();
    var question_numbers;
```

```
  var count = 0;
  var message;


  for (var ix=FIRST_QUESTION_ELEMENT; ix <= FIRST_QUESTION_ELEMENT+MAX_QUESTIONS-1;ix++){
    if (document.survey.elements[ix].selectedIndex == 0){
      errors[count++] = (ix - (FIRST_QUESTION_ELEMENT - 1));
      problem = true;
    }
  }

  if (problem){
    question_numbers = errors.join(", ");
    message = "The following question(s) was/were not answered: ";
    message += question_numbers;
    error_messages[error_count++] = message;
    form_errors = true;
  }
}//end check_survey_questions()



//-----------------------------------------------------------------------------
//Function:    reset()
//Parameters:  None
//Return Value: None
//Purpose:     Resets the global variable values for form_errors and error_count.
//-----------------------------------------------------------------------------
function reset(){

  form_errors = false;
  error_count = 0;
  error_messages = new Array();
}//end reset()



//-----------------------------------------------------------------------------
//Function:    check_form()
//Parameters:  None
//Return Value: None
//Purpose:     Calls utility functions to aid in the checking of form data.  If a problem
//             is found, the user is notified, and the form is not submitted until the
//             problem is corrected.
//-----------------------------------------------------------------------------
function check_form(){

  var msg;
  check_bio_information();
  check_survey_questions();
  if (form_errors){
    if (error_messages.length > 1){
      alert(error_messages[0] + "\n\n" + error_messages[1]);
    }
    else alert(error_messages[0]);
    reset();
    return false;
  }
```

```
  //get the user's survey_id and squadron_number if there are no problems
  get_squadron_number();
  get_survey_id();
  reset();
  return true;
}//end check_form()


//---------------------------------------------------------------------------------
//Function:    get_survey_id()
//Parameters:  None
//Return Value: None
//Purpose:     Queries the client for his/her survey_id number.
//---------------------------------------------------------------------------------
function get_survey_id(){

   var survey_id;
   do {
     survey_id = new String(prompt("Please enter the survey id you were assigned (Valid characters 0-9 only):", ""));
   } while (!(check_string(survey_id)));
     document.survey.survey_number.value = survey_id;
}//end get_survey_id()


//---------------------------------------------------------------------------------
//Function:    get_squadron_number()
//Parameters:  None
//Return Value: None
//Purpose:     Queries the client for his/her squadron number.
//---------------------------------------------------------------------------------
function get_squadron_number(){

  var squadron_num;
  do {
    squadron_num = new String(prompt("Please enter your squadron number (Valid characters 0-9 only):", ""));
  } while (!(check_string(squadron_num)));
    document.survey.squadron_number.value = squadron_num;
}//end get_squadron_number()
```

APPENDIX F. SERVLET SOURCE CODE

A.   SURVEYSERVLET.JAVA

```java
//packages we will use that need to be imported
import java.io.*;
import java.sql.*;
import javax.servlet.*;
import javax.servlet.http.*;
import file_utilities.*;

public class SurveyServlet extends HttpServlet
{
    final int NUM_DATA_ITEMS = 70;
    final int NUM_BIO_QUESTIONS = 9;
    final int NUM_SURVEY_QUESTIONS = 61;

    java.sql.Connection con1, con2;
    java.sql.Statement stmt1;
    java.sql.PreparedStatement stmt2;
    java.lang.String ipAddress, host, squadron_type, unit_number, survey_number;

    //ODBC data source name is survey, which references the Survey.mdb file
    java.lang.String url1 = "jdbc:ff-microsoft://spitfire.avsafety.nps.navy.mil:1433/Survey2_DB";
    java.lang.String url2 = "jdbc:ff-microsoft://spitfire.avsafety.nps.navy.mil:1433/Num_List_DB";

    java.lang.String[] clientdata = new java.lang.String[NUM_DATA_ITEMS];

    boolean entryError = false, generalError = false;

    //worker thread to write the client's data to a backup file
    file_utilities.WriteToFile fileop;


    //-----------------------------------------------------------------------
    //Function:   checkSurveyNumbersDatabase()
    //Parameters: None
    //Return type: boolean, indicating whether the survey number is good or not
    //Purpose:    Determines if the survey number corresponds to a legitimate
    //            entry for the given squadron.
    //-----------------------------------------------------------------------
    private boolean checkSurveyNumbersDatabase(){

        boolean valid = false;
        int count = 0, start = -1, end = -1;

        try {
            stmt2.setString(1, squadron_type+unit_number);
            java.sql.ResultSet rs = stmt2.executeQuery();

            if (rs != null){
                while (rs.next()){
                    rs.getString(1);
                    start = rs.getInt(2);
```

```java
      end  = rs.getInt(3);
      System.out.println("Start is " + start + " End is " + end);
   }

   if ((Integer.parseInt(survey_number) >= start) && (Integer.parseInt(survey_number) <= end)){
      System.out.println("We have a valid survey number!");
      valid = true;
   }
   else {
      System.out.println("We have an invalid survey number!");
   }
}
else {
   System.out.println("We have an invalid survey number!");
}
rs.close();
}
catch (java.sql.SQLException e){
   System.out.println(e.toString());
}
return valid;
}//end checkSurveyNumbersDatabase()


//--------------------------------------------------------------------------
//Function:   destroy()
//Parameters:  None
//Return type: None
//Purpose:    Frees any shared servlet resources
//            Called when the servletrunner program is terminated
//--------------------------------------------------------------------------
public void destroy() {
   //close the connection to the database - underlying code takes care of
   //actually releasing the connection
   try {
      stmt1.close();
      stmt2.close();
      con1.close();
      con2.close();
   }
   catch (java.sql.SQLException e){
      System.out.println(e.toString());
   }
}//end destroy()
```

```
//----------------------------------------------------------------------
//Function:   doPost()
//Parameters: request - an HttpServletRequest object that allows us to
//                      grab the values the client sent the servlet
//            response - an HttpServletResponse object that allows us to
//                       return a response back to the client
//Return type: None
//Purpose:    Makes several calls to utility functions that operate on the
//            client's input - the client is redirected to a different page
//            depending on the success or failure of the POST operation.
//----------------------------------------------------------------------
public void doPost(HttpServletRequest request, HttpServletResponse response)
  throws ServletException, IOException {

  entryError = false;
  boolean goodID = false;

  if (!generalError){
    parseFormData(request);

    goodID = checkSurveyNumbersDatabase();
    if (goodID){
      makeDatabaseEntry(response);

      if (!entryError){
        fileop.start();
        fileop.write(clientdata, ipAddress, host);
        try {
          response.setContentType("text/html");
          response.sendRedirect("http://spitfire.avsafety.nps.navy.mil/ThankYou.html");
        }
        catch (java.io.IOException e){
          System.out.println(e.toString());
        }
      }
    }
    else {
      response.sendRedirect("http://spitfire.avsafety.nps.navy.mil/NumberError.html");
    }
  }
  else
    response.sendRedirect("http://spitfire.avsafety.nps.navy.mil/GeneralError.html");

}//end doPost()
```

```
//-------------------------------------------------------------------------
//Function:  handleSQLException()
//Parameters: response - an HttpServletResponse object that allows us to
//                    return a response back to the client
//          ex - an SQLException
//Return type: None
//Purpose:    Generates HTML code that indicates to the user what SQL Error
//          was generated.
//-------------------------------------------------------------------------
private void handleSQLException(HttpServletResponse response, java.sql.SQLException ex){

    java.lang.String errorMessage = new String();
    while (ex != null){
      errorMessage += ("<br>Message: " + ex.getMessage() + "<br>SQL State: " + ex.getSQLState()
            + "<br>Error Code: " + ex.getErrorCode() + "<br>");
      System.out.println("Error executing SQL statements. Transaction not entered.\n");
      System.out.println("----------------------------------------------------");
      ex = ex.getNextException();
    }

    try {
      PrintWriter out;
      response.setContentType("text/html");
      out = response.getWriter();
      out.println("<html><head><title>Error!</title></head><body bgcolor=#000000 link=#00FFFF>" +
            "<p><font color=#FFFF00 size=4>A problem was encountered while " +
            "trying to process your input. Possible causes for this are listed " +
            "below. If you have additional questions as to why your input was " +
            "not accepted, you may contact the webmaster at </font>" +
            "<big><a HREF=mailto:spitfire.avsafety.nps.navy.mil>AVSafetyCenter</a></big>" +
            "<font color=#FFFF00 size=4>, or you may complete the survey again " +
            "and try resubmitting it.</font></p><p><font color=#FF0000 size=4> " +
            errorMessage + "</big></big></font></p></body></html>");
    }
    catch (java.io.IOException e){
      System.out.println(e.toString());
    }
}//end handleSQLException()


//-------------------------------------------------------------------------
//Function:  init()
//Parameters:  config - a servlet configuration object; this object is not
//                    manipulated in the code; rather, the superclass is called,
//                    and the object is manipulated by that class
//Return type: None
//Purpose:    Initializes the servlet. Resource-intensive procedures, such as
//          making a database connection, as well as shared resources, such
//          as the WriteToFile object, which stores a backup of the survey
//          information, is done here.
//-------------------------------------------------------------------------
public void init(ServletConfig config) throws ServletException {
    super.init(config);

    //shared resources of all servlets - a connection to the database, and
    //a thread that saves a backup of form data to a plaintext file in the
```

216

```
    //logs directory
    try {
      Class.forName("connect.microsoft.MicrosoftDriver");
      con1 = DriverManager.getConnection(url1, "sa", "");
      con2 = DriverManager.getConnection(url2, "sa", "");
      stmt1 = con1.createStatement();
      stmt2 = con2.prepareStatement("SELECT * FROM Num_List WHERE Unit_ID = ?");
      fileop = new file_utilities.WriteToFile();
    }
    catch (java.lang.ClassNotFoundException e) {
      System.out.println(e.toString());
      generalError = true;
    }
    catch (java.sql.SQLException e){
      generalError = true;
    }
}//end init()



//------------------------------------------------------------------------
//Function:   makeDatabaseEntry()
//Parameters: response - an HttpServletResponse object that is passed onto
//                 the method handleSQLException, if an SQLException is
//                 generated in the process of entering client information
//Return type: None
//Purpose:    Enters the clients data into the database via JDBC calls.
//------------------------------------------------------------------------
private void makeDatabaseEntry(HttpServletResponse response){
  try {

    java.lang.String operation = "INSERT INTO Biographical_Data VALUES('" +
              clientdata[0] + "'," +
              Integer.parseInt(unit_number) + "," +
              Integer.parseInt(survey_number) + "," +
              clientdata[1] + ",'" +
              clientdata[2] + "'," +
              Integer.parseInt(clientdata[3]) + "," +
              Integer.parseInt(clientdata[4]) + ",'" +
              clientdata[5] + "','" +
              clientdata[6] + "','" +
              clientdata[7] + "','" +
              clientdata[8] + "')";
    System.out.println("\nReceived client data. Parsing into database.");
    System.out.println("Executing SQL statements:\n");
    System.out.println(operation);
    stmt1.executeUpdate(operation);

    operation = "INSERT INTO Questions VALUES('" +
              clientdata[0] + "'," +
              Integer.parseInt(unit_number) + "," +
              Integer.parseInt(survey_number) + ",";

    for (int ix=NUM_BIO_QUESTIONS; ix<NUM_DATA_ITEMS-1; ix++)
      operation += (clientdata[ix] + ",");
    operation += clientdata[NUM_DATA_ITEMS-1] + ")";
    System.out.println(operation);
```

```java
        stmt1.executeUpdate(operation);
        System.out.println("SQL statements executed successfully.");
    }
    catch (java.sql.SQLException e){
        entryError = true;
        handleSQLException(response, e);
    }
}//end makeDatabaseEntry()


//-------------------------------------------------------------------------
//Function:   parseFormData()
//Parameters: request - an HttpServletRequest object that allows us to
//                      grab the values the client sent the servlet
//Return type: None
//Purpose:    Retrieves the form information that was sent to the servlet.
//-------------------------------------------------------------------------
private void parseFormData(HttpServletRequest request) {

    ipAddress = request.getRemoteAddr();
    host = request.getRemoteHost();

    //form parameters - form is made up of bio_data and questions
    squadron_type = new String(request.getParameter("community"));
    unit_number = new String(request.getParameter("squadron_number"));
    survey_number = new String(request.getParameter("survey_number"));

    java.lang.String bio_data[] = {"community", "rank", "designation", "flight_hours", "type_hours",
                    "department_head", "status", "service", "location"};

    java.lang.String questions[] = new String[NUM_SURVEY_QUESTIONS];

    for (int ix=0; ix<NUM_SURVEY_QUESTIONS; ix++)
        questions[ix] = "question" + Integer.toString(ix+1);

    for (int jx=0; jx < clientdata.length; jx++){
        if (jx < NUM_BIO_QUESTIONS)
            clientdata[jx] = request.getParameter(bio_data[jx]);
        else
            clientdata[jx] = request.getParameter(questions[jx-NUM_BIO_QUESTIONS]);
    }

} //end parseFormData()
}//end class SurveyServlet
```

B.   **WRITETOFILE.JAVA**

```java
package file_utilities;
import java.io.*;

public class WriteToFile extends Thread {
    static final java.lang.String resultsFile = "\\Inetpub\\wwwroot\\servlets\\survey.txt";
    static final java.io.FileWriter fw;
    static final int DATA_ITEMS = 70;

    public WriteToFile(){
        //open a file to save survey results as a backup txt file
        try {
            fw = new FileWriter(resultsFile, true);
        }
        catch (java.io.IOException ex){
            System.out.println("Please ensure the directory servlets exists!");
            System.out.println("Otherwise, backup survey data will not be kept.");
        }
    }

    public void write(java.lang.String data[], java.lang.String ipAddress,
                java.lang.String host) throws IOException {

        java.io.PrintWriter toFile;

        synchronized (fw){
            toFile = new PrintWriter(fw);
            java.util.Calendar ts = java.util.Calendar.getInstance();

            System.out.println("Submission entered for:\n" + ipAddress + "/" + host +"\n");
            System.out.println("----------------------------------------------");
            toFile.println("<FROM " + ipAddress + "/" + host + ">");
            toFile.println("<RECEIVED ON " + ts.getTime().toString() + ">");
            toFile.println("<BIO DATA>");
            for (int ix=0; ix<DATA_ITEMS; ix++){
                if (ix == 10){
                    toFile.println("<SURVEY DATA>");
                }
                if ((ix!=0) && (ix%9 == 0))
                    toFile.println(data[ix] + " ");
                else
                    toFile.print(data[ix] + " ");
            }
            toFile.println("");
            toFile.println("----------------------------------------------");
            fw.close();
        }
    }
}
```

A.   *SURVEY.JAVA* SOURCE CODE

```java
import java.awt.*;
import java.applet.*;
import Questionnaire;

public class Survey extends Applet
{
    boolean displayed = false;

        public void init()
        {
                setLayout(null);
                setSize(650,130);
                setBackground(new Color(0));
                //}}
        }

        public void paint(Graphics g){
            g.setColor(Color.yellow);
            g.setFont(new Font("Serif", Font.PLAIN, 30));
        g.drawString("Please wait while the program loads...", 80,80);

         if (!displayed){
                Questionnaire q = new Questionnaire();
            q.show(this);
            displayed = true;
         }
     }
}
```

B.   QUESTIONNAIRE.JAVA SOURCE CODE

```java
import java.awt.*;
import java.sql.*;
import java.net.*;
import symantec.itools.awt.TabPanel;
import symantec.itools.awt.BorderPanel;
import symantec.itools.multimedia.NervousText;
import symantec.itools.awt.WrappingLabel;
import symantec.itools.awt.ImagePanel;
import symantec.itools.awt.TabPanel;
import symantec.itools.awt.BorderPanel;
import UserInput;
import QuitDialog;

public class Questionnaire extends Frame
{
        public Questionnaire()
        {

        //{{INIT_CONTROLS
```

221

```java
setLayout(null);
setVisible(false);
setSize(insets().left + insets().right + 910,insets().top + insets().bottom + 579);
setResizable(false);
setBackground(new Color(128,128,128));

tabPanel1 = new symantec.itools.awt.TabPanel();
try {
    java.lang.String[] tempString = {"Bio", "Questions 1-9", "Questions 10-18",
                                     "Questions 19-28", "Questions 29-40", "Questions 41-52",
                                     "Questions 53-61", "Submit Survey!"};
    tabPanel1.setPanelLabels(tempString);
    tabPanel1.setCurrentPanelNdx(0);
}
catch(java.beans.PropertyVetoException e) { }

for (int ix=0; ix<NUM_QUESTIONS; ix++){
    choices[ix] = new java.awt.Choice();
        choices[ix].addItem("Strongly Disagree");
        choices[ix].addItem("Moderately Disagree");
        choices[ix].addItem("Slightly Disagree");
        choices[ix].addItem("NEUTRAL");
        choices[ix].addItem("Slightly Agree");
        choices[ix].addItem("Moderately Agree");
        choices[ix].addItem("Strongly Agree");
        choices[ix].addItem("NOT APPLICABLE");
        choices[ix].setBackground(new Color(16777215));

        try {
          choices[ix].select(-1);
        } catch (IllegalArgumentException e) { }

    wrappingLabels[ix] = new symantec.itools.awt.WrappingLabel();
    wrappingLabels[ix].setFont(new Font("Dialog", Font.PLAIN, 14));
    if (ix%2 != 0)
        wrappingLabels[ix].setForeground(new Color(16777215));
    else wrappingLabels[ix].setForeground(new Color(0));
}

    tabPanel1.setBounds(0,0,910,579);
    tabPanel1.setBackground(new Color(12632256));
    add(tabPanel1);
    borderPanel1 = new symantec.itools.awt.BorderPanel();
    borderPanel1.setLayout(null);
    borderPanel1.setVisible(false);
    borderPanel1.setBounds(12,33,888,520);
    borderPanel1.setBackground(new Color(-10707037));
    tabPanel1.add(borderPanel1);
    rankLabel = new java.awt.Label("Your rank:",Label.RIGHT);
    rankLabel.setBounds(158,63,180,26);
    rankLabel.setFont(new Font("Dialog", Font.PLAIN, 18));
    rankLabel.setForeground(new Color(0));
    borderPanel1.add(rankLabel);
    designationLabel = new java.awt.Label("Your designation:",Label.RIGHT);
    designationLabel.setBounds(158,99,180,26);
    designationLabel.setFont(new Font("Dialog", Font.PLAIN, 18));
```

222

```java
designationLabel.setForeground(new Color(16777215));
borderPanel1.add(designationLabel);
communityLabel = new java.awt.Label("Your community:",Label.RIGHT);
communityLabel.setBounds(158,135,180,26);
communityLabel.setFont(new Font("Dialog", Font.PLAIN, 18));
communityLabel.setForeground(new Color(0));
borderPanel1.add(communityLabel);
flightHoursLabel = new java.awt.Label("Total flight hours:",Label.RIGHT);
flightHoursLabel.setBounds(158,171,180,26);
flightHoursLabel.setFont(new Font("Dialog", Font.PLAIN, 18));
flightHoursLabel.setForeground(new Color(16777215));
borderPanel1.add(flightHoursLabel);
typeHoursLabel = new java.awt.Label("Hours in type:",Label.RIGHT);
typeHoursLabel.setBounds(158,207,180,26);
typeHoursLabel.setFont(new Font("Dialog", Font.PLAIN, 18));
typeHoursLabel.setForeground(new Color(0));
borderPanel1.add(typeHoursLabel);
DHLabel = new java.awt.Label("Are you a department head?",Label.RIGHT);
DHLabel.setBounds(86,243,252,26);
DHLabel.setFont(new Font("Dialog", Font.PLAIN, 18));
DHLabel.setForeground(new Color(16777215));
borderPanel1.add(DHLabel);
statusLabel = new java.awt.Label("Your status:",Label.RIGHT);
statusLabel.setBounds(110,279,228,26);
statusLabel.setFont(new Font("Dialog", Font.PLAIN, 18));
statusLabel.setForeground(new Color(0));
borderPanel1.add(statusLabel);
serviceLabel = new java.awt.Label("Your service:",Label.RIGHT);
serviceLabel.setBounds(110,315,228,26);
serviceLabel.setFont(new Font("Dialog", Font.PLAIN, 18));
serviceLabel.setForeground(new Color(16777215));
borderPanel1.add(serviceLabel);
locationLabel = new java.awt.Label("Your unit's location:",Label.RIGHT);
locationLabel.setBounds(158,351,180,26);
locationLabel.setFont(new Font("Dialog", Font.PLAIN, 18));
locationLabel.setForeground(new Color(0));
borderPanel1.add(locationLabel);
rankChoice = new java.awt.Choice();
rankChoice.addItem("O1-O3");
rankChoice.addItem("O4-O5");
rankChoice.addItem("O6-O10");
rankChoice.addItem("CWO1-CWO5");
try {
  rankChoice.select(-1);
} catch (IllegalArgumentException e) { }
borderPanel1.add(rankChoice);
rankChoice.setBounds(350,63,158,27);
rankChoice.setForeground(new Color(0));
rankChoice.setBackground(new Color(16777215));
designationChoice = new java.awt.Choice();
designationChoice.addItem("Pilot");
designationChoice.addItem("NFO");
designationChoice.addItem("Other");
try {
  designationChoice.select(-1);
} catch (IllegalArgumentException e) { }
```

```java
borderPanel1.add(designationChoice);
designationChoice.setBounds(350,99,158,27);
designationChoice.setForeground(new Color(0));
designationChoice.setBackground(new Color(16777215));
communityChoice = new java.awt.Choice();
communityChoice.addItem("HC");
communityChoice.addItem("HCS");
communityChoice.addItem("HM");
communityChoice.addItem("HMH");
communityChoice.addItem("HMT");
communityChoice.addItem("HS");
communityChoice.addItem("HSC");
communityChoice.addItem("HSL");
communityChoice.addItem("VAQ");
communityChoice.addItem("VAW");
communityChoice.addItem("VF");
communityChoice.addItem("VFA");
communityChoice.addItem("VMAQ");
communityChoice.addItem("VMFA");
communityChoice.addItem("VP");
communityChoice.addItem("VQ");
communityChoice.addItem("VR");
communityChoice.addItem("VRC");
communityChoice.addItem("VS");
communityChoice.addItem("VT");
communityChoice.addItem("VX");
communityChoice.addItem("Other");
try {
    communityChoice.select(-1);
} catch (IllegalArgumentException e) { }
borderPanel1.add(communityChoice);
communityChoice.setBounds(350,135,158,27);
communityChoice.setForeground(new Color(0));
communityChoice.setBackground(new Color(16777215));
flightHoursText = new java.awt.TextField();
flightHoursText.setBounds(350,171,156,23);
flightHoursText.setFont(new Font("Dialog", Font.PLAIN, 18));
flightHoursText.setBackground(new Color(16777215));
borderPanel1.add(flightHoursText);
typeHoursText = new java.awt.TextField();
typeHoursText.setBounds(350,207,156,23);
typeHoursText.setFont(new Font("Dialog", Font.PLAIN, 18));
typeHoursText.setBackground(new Color(16777215));
borderPanel1.add(typeHoursText);
DepartmentHead = new CheckboxGroup();
yesRadioButton = new java.awt.Checkbox("Yes", DepartmentHead, false);
yesRadioButton.setBounds(362,243,72,24);
yesRadioButton.setFont(new Font("Dialog", Font.PLAIN, 18));
yesRadioButton.setForeground(new Color(255));
yesRadioButton.setBackground(new Color(6070179));
borderPanel1.add(yesRadioButton);
noRadioButton = new java.awt.Checkbox("No", DepartmentHead, false);
noRadioButton.setBounds(494,243,48,24);
noRadioButton.setFont(new Font("Dialog", Font.PLAIN, 18));
noRadioButton.setForeground(new Color(255));
noRadioButton.setBackground(new Color(6070179));
```

```java
borderPanel1.add(noRadioButton);
status = new CheckboxGroup();
regularRadioButton = new java.awt.Checkbox("Regular", status, false);
regularRadioButton.setBounds(362,279,96,27);
regularRadioButton.setFont(new Font("Dialog", Font.PLAIN, 18));
regularRadioButton.setForeground(new Color(255));
regularRadioButton.setBackground(new Color(-10707037));
borderPanel1.add(regularRadioButton);
activeRadioButton = new java.awt.Checkbox("Active Reserve", status, false);
activeRadioButton.setBounds(494,279,144,27);
activeRadioButton.setFont(new Font("Dialog", Font.PLAIN, 18));
activeRadioButton.setForeground(new Color(255));
activeRadioButton.setBackground(new Color(-10707037));
borderPanel1.add(activeRadioButton);
drillingRadioButton = new java.awt.Checkbox("Drilling Reserve", status, false);
drillingRadioButton.setBounds(650,279,156,27);
drillingRadioButton.setFont(new Font("Dialog", Font.PLAIN, 18));
drillingRadioButton.setForeground(new Color(255));
drillingRadioButton.setBackground(new Color(-10707037));
borderPanel1.add(drillingRadioButton);
serviceChoice = new java.awt.Choice();
serviceChoice.addItem("USN");
serviceChoice.addItem("USMC");
serviceChoice.addItem("Other");
try {
  serviceChoice.select(-1);
} catch (IllegalArgumentException e) { }
borderPanel1.add(serviceChoice);
serviceChoice.setBounds(350,315,158,27);
serviceChoice.setForeground(new Color(0));
serviceChoice.setBackground(new Color(16777215));
location = new CheckboxGroup();
eastButton = new java.awt.Checkbox("East Coast", location, false);
eastButton.setBounds(362,351,120,27);
eastButton.setFont(new Font("Dialog", Font.PLAIN, 18));
eastButton.setForeground(new Color(255));
eastButton.setBackground(new Color(-10707037));
borderPanel1.add(eastButton);
westButton = new java.awt.Checkbox("West Coast", location, false);
westButton.setBounds(494,351,132,27);
westButton.setFont(new Font("Dialog", Font.PLAIN, 18));
westButton.setForeground(new Color(255));
westButton.setBackground(new Color(-10707037));
borderPanel1.add(westButton);
otherButton = new java.awt.Checkbox("Other", location, false);
otherButton.setBounds(650,351,78,27);
otherButton.setFont(new Font("Dialog", Font.PLAIN, 18));
otherButton.setForeground(new Color(255));
otherButton.setBackground(new Color(-10707037));
borderPanel1.add(otherButton);

//frame gifs

imagePanel1 = new symantec.itools.awt.ImagePanel();
imagePanel2 = new symantec.itools.awt.ImagePanel();
imagePanel3 = new symantec.itools.awt.ImagePanel();
```

```java
try {
  imagePanel1.setImageURL(new java.net.URL
        ("http://spitfire.avsafety.nps.navy.mil/Survey/images/Compass.gif"));
  imagePanel1.setStyle(symantec.itools.awt.ImagePanel.IMAGE_CENTERED);
  imagePanel1.setBounds(62,39,108,92);
  borderPanel1.add(imagePanel1);

  imagePanel2.setImageURL(new java.net.URL
        ("http://spitfire.avsafety.nps.navy.mil/Survey/images/Radar.gif"));
  imagePanel2.setStyle(symantec.itools.awt.ImagePanel.IMAGE_CENTERED);
  imagePanel2.setBounds(62,351,108,92);
  borderPanel1.add(imagePanel2);

  imagePanel3.setImageURL(new java.net.URL
        ("http://spitfire.avsafety.nps.navy.mil/Survey/images/fly.gif"));
  imagePanel3.setStyle(symantec.itools.awt.ImagePanel.IMAGE_SCALED_TO_FIT);
  imagePanel3.setBounds(564,75,276,192);
  borderPanel1.add(imagePanel3);
}
catch (java.net.MalformedURLException error) { }
catch(java.beans.PropertyVetoException e) { }

//beginning of second border panel, i.e. second tab

borderPanel2 = new symantec.itools.awt.BorderPanel();
borderPanel2.setLayout(null);
borderPanel2.setVisible(false);
borderPanel2.setBounds(12,33,888,520);
borderPanel2.setBackground(new Color(-10707037));
tabPanel1.add(borderPanel2);

nervousText1 = new symantec.itools.multimedia.NervousText();
try {
  nervousText1.setText("A. PROCESS");
}
catch(java.beans.PropertyVetoException e) { }
nervousText1.setBounds(2,3,216,48);
nervousText1.setFont(new Font("Helvetica", Font.BOLD, 36));
nervousText1.setForeground(new Color(16776960));
borderPanel2.add(nervousText1);

try {
  wrappingLabels[0].setText
        ("1. My command conducts adequate reviews and updates of safety standards " +
        "and operating procedures.");
  wrappingLabels[0].setBounds(2,63,672,20);
  borderPanel2.add(wrappingLabels[0]);

  wrappingLabels[1].setText
        ("2. My command uses an internal audit and hazard reporting system to catch " +
        "any problems that may lead to a mishap.");
  wrappingLabels[1].setBounds(2,111,672,48);
  borderPanel2.add(wrappingLabels[1]);

  wrappingLabels[2].setText
```

```
        ("3. My command has a defined process to set training goals and to review " +
            "performance.");
wrappingLabels[2].setBounds(2,171,564,20);
borderPanel2.add(wrappingLabels[2]);

wrappingLabels[3].setText
        ("4. My command closely monitors proficiency and currency standards to " +
            "ensure aircrew are qualified to fly.");
wrappingLabels[3].setBounds(2,219,672,20);
borderPanel2.add(wrappingLabels[3]);

wrappingLabels[4].setText
        ("5. Command leadership is actively involved in the safety program and " +
            "management of safety matters.");
wrappingLabels[4].setBounds(2,255,672,20);
borderPanel2.add(wrappingLabels[4]);

wrappingLabels[5].setText
        ("6. My command has a defined process to effectively manage the high-risk aviator.");
wrappingLabels[5].setBounds(2,303,672,20);
borderPanel2.add(wrappingLabels[5]);

wrappingLabels[6].setText
        ("7. Human Factors Councils have been successful in identifying aircrew " +
            "members who pose a risk to safety.");
wrappingLabels[6].setBounds(2,351,672,20);
borderPanel2.add(wrappingLabels[6]);

wrappingLabels[7].setText
        ("8. Human Factors Boards have been successful reducing chances of an " +
            "aircraft mishap due to the high-risk aviator.");
wrappingLabels[7].setBounds(2,399,660,32);
borderPanel2.add(wrappingLabels[7]);

wrappingLabels[8].setText
        ("9. My command makes effective use of the flight surgeon to help idenitfy " +
            "and manage high-risk personnel.");
wrappingLabels[8].setBounds(2,447,672,20);
borderPanel2.add(wrappingLabels[8]);
} catch(java.beans.PropertyVetoException e) { }


//add choice boxes to borderpanel2

borderPanel2.add(choices[0]);
choices[0].setBounds(686,63,172,21);
borderPanel2.add(choices[1]);
choices[1].setBounds(686,111,172,21);
borderPanel2.add(choices[2]);
choices[2].setBounds(686,171,172,21);
borderPanel2.add(choices[3]);
choices[3].setBounds(686,219,172,21);
borderPanel2.add(choices[4]);
choices[4].setBounds(686,255,172,21);
borderPanel2.add(choices[5]);
choices[5].setBounds(686,303,172,21);
```

```
borderPanel2.add(choices[6]);
choices[6].setBounds(686,351,172,21);
borderPanel2.add(choices[7]);
choices[7].setBounds(686,399,172,21);
borderPanel2.add(choices[8]);
choices[8].setBounds(686,447,172,21);

//beginning of borderpanel3, i.e. third tab

borderPanel3 = new symantec.itools.awt.BorderPanel();
borderPanel3.setLayout(null);
borderPanel3.setVisible(false);
borderPanel3.setBounds(12,33,888,520);
borderPanel3.setBackground(new Color(-10707037));
tabPanel1.add(borderPanel3);
nervousText2 = new symantec.itools.multimedia.NervousText();
try {
    nervousText2.setText("B. REWARD SYSTEM AND SAFETY CULTURE");
}
catch(java.beans.PropertyVetoException e) { }
nervousText2.setBounds(2,3,744,48);
nervousText2.setFont(new Font("Helvetica", Font.BOLD, 36));
nervousText2.setForeground(new Color(16776960));
borderPanel3.add(nervousText2);

try {
    wrappingLabels[9].setText
        ("10. Command leadership encourages reporting safety discrepancies without " +
        "the fear of negative repercussions.");
    wrappingLabels[9].setBounds(2,63,636,43);
    borderPanel3.add(wrappingLabels[9]);

    wrappingLabels[10].setText
        ("11. Individuals in my command are willing to report safety violations, " +
        "unsafe behaviors or hazardous conditions.");
    wrappingLabels[10].setBounds(2,111,672,48);
    borderPanel3.add(wrappingLabels[10]);

    wrappingLabels[11].setText
        ("12. In my command, peer influence is effective at discouraging violations " +
        "of standard operating procedures, or safety rules.");
    wrappingLabels[11].setBounds(2,171,664,36);
    borderPanel3.add(wrappingLabels[11]);

    wrappingLabels[12].setText
        ("13. In my command, we believe safety is an integral part of all flight " +
        "operations.");
    wrappingLabels[12].setBounds(2,219,672,20);
    borderPanel3.add(wrappingLabels[12]);

    wrappingLabels[13].setText
        ("14. In my command, anyone who intentionally violates standard procedures, " +
        "or safety rules, is swiftly corrected.");
    wrappingLabels[13].setBounds(2,255,675,36);
    borderPanel3.add(wrappingLabels[13]);
```

```java
wrappingLabels[14].setText
        ("15. In my command, violations of operating procedures, flying regulations, " +
        "or general flight discipline are rare.");
wrappingLabels[14].setBounds(2,303,675,36);
borderPanel3.add(wrappingLabels[14]);

wrappingLabels[15].setText
        ("16. Leaders in my command encourage everyone to be safety conscious and to " +
        "follow the rules.");
wrappingLabels[15].setBounds(2,351,672,20);
borderPanel3.add(wrappingLabels[15]);

wrappingLabels[16].setText
        ("17. In this command, an aviator who persistently violates flight standards " +
        "and rules will seriously jeopardize his/her career.");
wrappingLabels[16].setBounds(2,399,660,32);
borderPanel3.add(wrappingLabels[16]);
wrappingLabels[17].setText
        ("18. I am not comfortable reporting a safety violation, because people in " +
        "my command would react negatively toward me.");
wrappingLabels[17].setBounds(2,447,665,36);
borderPanel3.add(wrappingLabels[17]);
} catch(java.beans.PropertyVetoException e) { }


//add choice boxes to borderpanel3

borderPanel3.add(choices[9]);
choices[9].setBounds(686,63,172,21);
borderPanel3.add(choices[10]);
choices[10].setBounds(686,111,172,21);
borderPanel3.add(choices[11]);
choices[11].setBounds(686,171,172,21);
borderPanel3.add(choices[12]);
choices[12].setBounds(686,219,172,21);
borderPanel3.add(choices[13]);
choices[13].setBounds(686,255,172,21);
borderPanel3.add(choices[14]);
choices[14].setBounds(686,303,172,21);
borderPanel3.add(choices[15]);
choices[15].setBounds(686,351,172,21);
borderPanel3.add(choices[16]);
choices[16].setBounds(686,399,172,21);
borderPanel3.add(choices[17]);
choices[17].setBounds(686,447,172,21);

//beginning of borderpanel4, i.e. fourth tab

borderPanel4 = new symantec.itools.awt.BorderPanel();
borderPanel4.setLayout(null);
borderPanel4.setVisible(false);
borderPanel4.setBounds(12,33,888,520);
borderPanel4.setBackground(new Color(-10707037));
tabPanel1.add(borderPanel4);
nervousText3 = new symantec.itools.multimedia.NervousText();
try {
```

```java
        nervousText3.setText("C. QUALITY");
}
catch(java.beans.PropertyVetoException e) { }
nervousText3.setBounds(2,3,228,48);
nervousText3.setFont(new Font("Helvetica", Font.BOLD, 36));
nervousText3.setForeground(new Color(16776960));
borderPanel4.add(nervousText3);

try {
    wrappingLabels[18].setText
            ("19. My command has a reputation for high quality performance.");
    wrappingLabels[18].setBounds(2,63,408,24);
    borderPanel4.add(wrappingLabels[18]);

    wrappingLabels[19].setText
            ("20. My command sets high quality standards and strives to maintain " +
            "quality control.");
    wrappingLabels[19].setBounds(2,99,540,24);
    borderPanel4.add(wrappingLabels[19]);

    wrappingLabels[20].setText
            ("21. My command closely monitors quality and corrects any deviations from " +
            "established quality standards.");
    wrappingLabels[20].setBounds(2,135,664,36);
    borderPanel4.add(wrappingLabels[20]);

    wrappingLabels[21].setText
            ("22. Quality standards in my command are clearly stated in formal " +
            "publications and procedural guides.");
    wrappingLabels[21].setBounds(2,171,672,20);
    borderPanel4.add(wrappingLabels[21]);

    nervousText4 = new symantec.itools.multimedia.NervousText();
    nervousText4.setText("D. RISK MANAGEMENT");
    nervousText4.setBounds(2,195,396,48);
    nervousText4.setFont(new Font("Helvetica", Font.BOLD, 36));
    nervousText4.setForeground(new Color(16776960));
    borderPanel4.add(nervousText4);

    wrappingLabels[22].setText("23. Command leaders permit cutting corners to get a job done.");
    wrappingLabels[22].setBounds(2,267,408,24);
    borderPanel4.add(wrappingLabels[22]);

    wrappingLabels[23].setText
            ("24. Lack of experienced personnel has adversely affected my command's " +
            "ability to operate safely.");
    wrappingLabels[23].setBounds(2,303,648,24);
    borderPanel4.add(wrappingLabels[23]);

    wrappingLabels[24].setText
            ("25. Safety decisions are made at the proper levels, by the most " +
            "qualified people in my command.");
    wrappingLabels[24].setBounds(2,339,664,36);
    borderPanel4.add(wrappingLabels[24]);

    wrappingLabels[25].setText
```

```java
        ("26. Command leaders consider safety issues during the formation and " +
        "execution of operational and training plans.");
wrappingLabels[25].setBounds(2,375,648,36);
borderPanel4.add(wrappingLabels[25]);

wrappingLabels[26].setText
        ("27. Command leadership has a clear picture of the risks associated " +
        "with its flight operations.");
wrappingLabels[26].setBounds(2,423,612,24);
borderPanel4.add(wrappingLabels[26]);

wrappingLabels[27].setText
        ("28. My command takes the time to identify and assess risks associated " +
        "with its flight operations.");
wrappingLabels[27].setBounds(2,459,612,24);
borderPanel4.add(wrappingLabels[27]);
} catch(java.beans.PropertyVetoException e) { }

//add choice boxes to borderpanel4

borderPanel4.add(choices[18]);
choices[18].setBounds(686,63,172,21);
borderPanel4.add(choices[19]);
choices[19].setBounds(686,99,172,21);
borderPanel4.add(choices[20]);
choices[20].setBounds(686,135,172,21);
borderPanel4.add(choices[21]);
choices[21].setBounds(686,171,172,21);
borderPanel4.add(choices[22]);
choices[22].setBounds(686,267,172,21);
borderPanel4.add(choices[23]);
choices[23].setBounds(686,303,172,21);
borderPanel4.add(choices[24]);
choices[24].setBounds(686,339,172,21);
borderPanel4.add(choices[25]);
choices[25].setBounds(686,375,172,21);
borderPanel4.add(choices[26]);
choices[26].setBounds(686,423,172,21);
borderPanel4.add(choices[27]);
choices[27].setBounds(686,459,172,21);


//beginning of borderpanel5, i.e. fifth tab

borderPanel5 = new symantec.itools.awt.BorderPanel();
borderPanel5.setLayout(null);
borderPanel5.setVisible(false);
borderPanel5.setBounds(12,33,888,520);
borderPanel5.setBackground(new Color(-10707037));
tabPanel1.add(borderPanel5);

try {
  wrappingLabels[28].setText
        ("29. My command does a good job managing risks associated with its " +
        "flight operations.");
  wrappingLabels[28].setBounds(2,27,552,24);
```

231

```java
borderPanel5.add(wrappingLabels[28]);

wrappingLabels[29].setText
        ("30. My command has increased the chances of a mishap due to inadequate " +
        "or incorrect risk assessment.");
wrappingLabels[29].setBounds(2,63,648,36);
borderPanel5.add(wrappingLabels[29]);

wrappingLabels[30].setText
        ("31. I am provided adequate resources (time, staffing, budget, and equipment) " +
        "to accomplish my job.");
wrappingLabels[30].setBounds(2,111,664,36);
borderPanel5.add(wrappingLabels[30]);

wrappingLabels[31].setText
        ("32. My command provides the right number of flight hours per month for " +
        "me to fly safely.");
wrappingLabels[31].setBounds(2,147,648,36);
borderPanel5.add(wrappingLabels[31]);

wrappingLabels[32].setText
        ("33. I have adequate time to prepare for and brief my flights.");
wrappingLabels[32].setBounds(2,195,612,24);
borderPanel5.add(wrappingLabels[32]);

wrappingLabels[33].setText
        ("34. Based upon my command's personnel and other assets, the command " +
        "is over-committed.");
wrappingLabels[33].setBounds(2,231,612,24);
borderPanel5.add(wrappingLabels[33]);

wrappingLabels[34].setText
        ("35. My command has incorporated Operational Risk Management processes " +
        "in decision-making at all levels.");
wrappingLabels[34].setBounds(2,267,660,36);
borderPanel5.add(wrappingLabels[34]);

wrappingLabels[35].setText("36. My supervisor can be relied on to keep his/her word.");
wrappingLabels[35].setBounds(2,315,648,24);
borderPanel5.add(wrappingLabels[35]);

wrappingLabels[36].setText("37. Our command leaders and supervisors can be trusted.");
wrappingLabels[36].setBounds(2,351,664,24);
borderPanel5.add(wrappingLabels[36]);

wrappingLabels[37].setText("38.My command's Safety Officer is highly regarded.");
wrappingLabels[37].setBounds(2,387,648,24);
borderPanel5.add(wrappingLabels[37]);

wrappingLabels[38].setText("39. Our Safety Officer is influential in promoting safety.");
wrappingLabels[38].setBounds(2,423,612,24);
borderPanel5.add(wrappingLabels[38]);

wrappingLabels[39].setText("40. My command is genuinely concerned about safety.");
wrappingLabels[39].setBounds(2,459,612,24);
borderPanel5.add(wrappingLabels[39]);
```

```java
} catch(java.beans.PropertyVetoException e) { }


//add choice boxes to borderpanel5

borderPanel5.add(choices[28]);
choices[28].setBounds(686,27,172,21);
borderPanel5.add(choices[29]);
choices[29].setBounds(686,63,172,21);
borderPanel5.add(choices[30]);
choices[30].setBounds(686,111,172,21);
borderPanel5.add(choices[31]);
choices[31].setBounds(686,147,172,21);
borderPanel5.add(choices[32]);
choices[32].setBounds(686,195,172,21);
borderPanel5.add(choices[33]);
choices[33].setBounds(686,231,172,21);
borderPanel5.add(choices[34]);
choices[34].setBounds(686,267,172,21);
borderPanel5.add(choices[35]);
choices[35].setBounds(686,315,172,21);
borderPanel5.add(choices[36]);
choices[36].setBounds(686,351,172,21);
borderPanel5.add(choices[37]);
choices[37].setBounds(686,387,172,21);
borderPanel5.add(choices[38]);
choices[38].setBounds(686,423,172,21);
borderPanel5.add(choices[39]);
choices[39].setBounds(686,459,172,21);

//beginning of borderpanel6, i.e. sixth tab

borderPanel6 = new symantec.itools.awt.BorderPanel();
borderPanel6.setLayout(null);
borderPanel6.setVisible(false);
borderPanel6.setBounds(12,33,888,520);
borderPanel6.setBackground(new Color(-10707037));
tabPanel1.add(borderPanel6);

try {
  wrappingLabels[40].setText
        ("41. Command leadership is successful in communicating its safety goals " +
        "to unit personnel.");
  wrappingLabels[40].setBounds(2,27,588,24);
  borderPanel6.add(wrappingLabels[40]);

  wrappingLabels[41].setText
        ("42. My command provides a positive command climate that promotes safe " +
        "flight operations.");
  wrappingLabels[41].setBounds(2,63,648,36);
  borderPanel6.add(wrappingLabels[41]);

  wrappingLabels[42].setText
        ("43. Command leadership is actively involved in the safety program and " +
        "management of safety matters.");
  wrappingLabels[42].setBounds(2,111,664,36);
```

```java
borderPanel6.add(wrappingLabels[42]);

wrappingLabels[43].setText
        ("44. Command leadership sets the example for compliance with flight standards.");
wrappingLabels[43].setBounds(2,147,648,36);
borderPanel6.add(wrappingLabels[43]);

wrappingLabels[44].setText
        ("45. My command ensures that all unit members are responsible and " +
        "accountable for safe flight operations.");
wrappingLabels[44].setBounds(2,195,684,24);
borderPanel6.add(wrappingLabels[44]);

wrappingLabels[45].setText
        ("46. Command leadership willingly assists in providing advice concerning " +
        "safety matters.");
wrappingLabels[45].setBounds(2,231,612,24);
borderPanel6.add(wrappingLabels[45]);

wrappingLabels[46].setText
        ("47. Command leadership reacts well to unexpected changes to its plans.");
wrappingLabels[46].setBounds(2,267,660,36);
borderPanel6.add(wrappingLabels[46]);

wrappingLabels[47].setText
        ("48. My command does not hesitate to temporarily restrict from flying " +
        "individuals who are under high personal stress.");
wrappingLabels[47].setBounds(2,315,648,36);
borderPanel6.add(wrappingLabels[47]);

wrappingLabels[48].setText("49. I am adequately trained to safely conduct all of my flights.");
wrappingLabels[48].setBounds(2,351,664,24);
borderPanel6.add(wrappingLabels[48]);

wrappingLabels[49].setText("50. Morale and motivation in my command are high.");
wrappingLabels[49].setBounds(2,387,648,24);
borderPanel6.add(wrappingLabels[49]);

wrappingLabels[50].setText
        ("51. My command ensures the uniform enforcement of all operating standards " +
        "among unit members.");
wrappingLabels[50].setBounds(2,423,660,24);
borderPanel6.add(wrappingLabels[50]);

wrappingLabels[51].setText("52. Crew rest standards are enforced in my command.");
wrappingLabels[51].setBounds(2,459,612,24);
borderPanel6.add(wrappingLabels[51]);
} catch(java.beans.PropertyVetoException e) { }

//add choice boxes to borderpanel6

borderPanel6.add(choices[40]);
choices[40].setBounds(686,27,172,21);
borderPanel6.add(choices[41]);
choices[41].setBounds(686,63,172,21);
borderPanel6.add(choices[42]);
```

```java
choices[42].setBounds(686,111,172,21);
borderPanel6.add(choices[43]);
choices[43].setBounds(686,147,172,21);
borderPanel6.add(choices[44]);
choices[44].setBounds(686,195,172,21);
borderPanel6.add(choices[45]);
choices[45].setBounds(686,231,172,21);
borderPanel6.add(choices[46]);
choices[46].setBounds(686,267,172,21);
borderPanel6.add(choices[47]);
choices[47].setBounds(686,315,172,21);
borderPanel6.add(choices[48]);
choices[48].setBounds(686,351,172,21);
borderPanel6.add(choices[49]);
choices[49].setBounds(686,387,172,21);
borderPanel6.add(choices[50]);
choices[50].setBounds(686,423,172,21);
borderPanel6.add(choices[51]);
choices[51].setBounds(686,459,172,21);

//beginning of borderpanel7, i.e. seventh tab

borderPanel7 = new symantec.itools.awt.BorderPanel();
borderPanel7.setLayout(null);
borderPanel7.setBounds(12,33,888,520);
borderPanel7.setBackground(new Color(-10707037));
tabPanel1.add(borderPanel7);

try {
  wrappingLabels[52].setText
        ("53. In my command, NATOPS tests and check rides are conducted as " +
        "intended, to candidly assess aircrew qualifications.");
  wrappingLabels[52].setBounds(2,27,588,36);
  borderPanel7.add(wrappingLabels[52]);

  wrappingLabels[53].setText
        ("54. My command provides adequate safety backups to catch possible " +
        "human errors during high-risk missions.");
  wrappingLabels[53].setBounds(2,75,648,36);
  borderPanel7.add(wrappingLabels[53]);

  wrappingLabels[54].setText
        ("55. Withim my command, good communications flow exists up and down " +
        "the chain of command.");
  wrappingLabels[54].setBounds(2,123,664,36);
  borderPanel7.add(wrappingLabels[54]);

  wrappingLabels[55].setText
        ("56. My command has good two-way communication with external commands.");
  wrappingLabels[55].setBounds(2,159,648,36);
  borderPanel7.add(wrappingLabels[55]);

  wrappingLabels[56].setText
        ("57. Safety education and training are adequate in my command.");
  wrappingLabels[56].setBounds(2,207,684,24);
  borderPanel7.add(wrappingLabels[56]);
```

```
wrappingLabels[57].setText
    ("58. The Safety Department is a well-respected element of my command.");
wrappingLabels[57].setBounds(2,243,612,24);
borderPanel7.add(wrappingLabels[57]);

wrappingLabels[58].setText
    ("59. The Aviation Safety Officer position is a sought-after billet in " +
    "my command.");
wrappingLabels[58].setBounds(2,279,660,36);
borderPanel7.add(wrappingLabels[58]);

wrappingLabels[59].setText
    ("60. My command's Safety Department keeps me well informed regarding " +
    "important safety information.");
wrappingLabels[59].setBounds(2,327,648,36);
borderPanel7.add(wrappingLabels[59]);

wrappingLabels[60].setText
    ("61. My command's Aircrew Coordination Training program is helping to " +
    "improve mission performance and safety.");
wrappingLabels[60].setBounds(2,363,664,36);
borderPanel7.add(wrappingLabels[60]);
}
catch(java.beans.PropertyVetoException e) { }

borderPanel7.add(choices[52]);
choices[52].setBounds(686,27,172,21);
borderPanel7.add(choices[53]);
choices[53].setBounds(686,75,172,21);
borderPanel7.add(choices[54]);
choices[54].setBounds(686,123,172,21);
borderPanel7.add(choices[55]);
choices[55].setBounds(686,159,172,21);
borderPanel7.add(choices[56]);
choices[56].setBounds(686,207,172,21);
borderPanel7.add(choices[57]);
choices[57].setBounds(686,243,172,21);
borderPanel7.add(choices[58]);
choices[58].setBounds(686,279,172,21);
borderPanel7.add(choices[59]);
choices[59].setBounds(686,327,172,21);
borderPanel7.add(choices[60]);
choices[60].setBounds(686,363,172,21);


//beginning of borderpanel8

borderPanel8 = new symantec.itools.awt.BorderPanel();
borderPanel8.setLayout(null);
borderPanel8.setVisible(false);
borderPanel8.setBounds(12,33,888,520);
borderPanel8.setBackground(new Color(-10707037));
tabPanel1.add(borderPanel8);
wrappingLabel1 = new symantec.itools.awt.WrappingLabel();
try {
```

```java
            wrappingLabel1.setText("When you are ready to submit your survey, press the submit" +
            "button. The textbox below will report any errors, and it will inform you when " +
            "a successful entry to the database has been made. After clicking the " +
            "submit button, you will be prompted for your squadron and survey number " +
            "so we can authenticate your response. Your answers remain confidential, " +
            "and no attempts will be made to identify individuals or organizations. " +
            "Thank you for your participation.");
}
catch(java.beans.PropertyVetoException e) { }
wrappingLabel1.setBounds(74,63,718,96);
wrappingLabel1.setFont(new Font("Dialog", Font.BOLD, 14));
wrappingLabel1.setForeground(new Color(16762880));
borderPanel8.add(wrappingLabel1);
submitButton = new java.awt.Button();
submitButton.setActionCommand("submit");
submitButton.setLabel("Submit!");
submitButton.setBounds(374,183,158,51);
submitButton.setForeground(new Color(255));
submitButton.setBackground(new Color(12632256));
submitButton.setFont(new Font("Dialog", Font.PLAIN, 18));
borderPanel8.add(submitButton);
responseText = new java.awt.TextArea("",0,0,TextArea.SCROLLBARS_VERTICAL_ONLY);
responseText.setForeground(new Color(16711680));
responseText.setBackground(new Color(16777215));
responseText.setFont(new Font("Dialog", Font.BOLD, 14));
responseText.setEditable(false);
responseText.setBounds(74,280,729,194);
borderPanel8.add(responseText);

setTitle("ACSA Survey");
//}}


//{{INIT_MENUS
mainMenuBar = new java.awt.MenuBar();
menu1 = new java.awt.Menu("File");
miNew = new java.awt.MenuItem("New");
menu1.add(miNew);
menu1.addSeparator();
miExit = new java.awt.MenuItem("Exit");
menu1.add(miExit);
mainMenuBar.add(menu1);
menu2 = new java.awt.Menu("Help");
mainMenuBar.setHelpMenu(menu2);
miAbout = new java.awt.MenuItem("About..");
menu2.add(miAbout);
mainMenuBar.add(menu2);
setMenuBar(mainMenuBar);
//$$ mainMenuBar.move(4,277);
//}}


//{{REGISTER_LISTENERS
SymWindow aSymWindow = new SymWindow();
this.addWindowListener(aSymWindow);
SymAction lSymAction = new SymAction();
miAbout.addActionListener(lSymAction);
miNew.addActionListener(lSymAction);
```

```java
    miExit.addActionListener(lSymAction);
    SymMouse aSymMouse = new SymMouse();
    submitButton.addMouseListener(aSymMouse);
    //}}
}

public Questionnaire(java.lang.String title)
{
    this();
    setTitle(title);
}

public synchronized void show(java.applet.Applet FrameParent)
{
    move(50, 50);
    super.show();
    parent = FrameParent;
}

static public void main(String args[])
{
    (new Questionnaire()).show();
}

public void addNotify()
{
    // Record the size of the window prior to calling parents addNotify.
    Dimension d = getSize();

        super.addNotify();

        if (fComponentsAdjusted)
                return;

        // Adjust components according to the insets
        setSize(insets().left + insets().right + d.width, insets().top + insets().bottom + d.height);
        Component components[] = getComponents();
        for (int i = 0; i < components.length; i++)
{

            Point p = components[i].getLocation();
            p.translate(insets().left, insets().top);
            components[i].setLocation(p);
        }
        fComponentsAdjusted = true;
}

// Used for addNotify check.
boolean fComponentsAdjusted = false;

//{{DECLARE_CONTROLS
java.awt.FileDialog openFileDialog1;
final int NUM_QUESTIONS = 61;
symantec.itools.awt.TabPanel tabPanel1;
symantec.itools.awt.BorderPanel borderPanel1, borderPanel2, borderPanel3, borderPanel4,
borderPanel5, borderPanel6, borderPanel7, borderPanel8;
```

```
symantec.itools.awt.ImagePanel imagePanel1, imagePanel2, imagePanel3;
symantec.itools.multimedia.NervousText nervousText1, nervousText2, nervousText3,
nervousText4;

java.awt.Choice choices[] = new java.awt.Choice[NUM_QUESTIONS];
symantec.itools.awt.WrappingLabel wrappingLabels[] = new
symantec.itools.awt.WrappingLabel[NUM_QUESTIONS];
symantec.itools.awt.WrappingLabel wrappingLabel1;
java.awt.TextArea responseText;
java.awt.Button submitButton;
java.awt.Label rankLabel, designationLabel, communityLabel, flightHoursLabel, typeHoursLabel,
DHLabel, statusLabel, serviceLabel, locationLabel;
java.awt.Choice rankChoice, designationChoice, communityChoice, serviceChoice;
java.awt.TextField flightHoursText, typeHoursText;
java.awt.Checkbox yesRadioButton, noRadioButton, regularRadioButton, activeRadioButton,
drillingRadioButton, eastButton, westButton, otherButton;
CheckboxGroup DepartmentHead, status, location;
 boolean processing = false;
 java.applet.Applet parent;
//}}

//{{DECLARE_MENUS
java.awt.MenuBar mainMenuBar;
java.awt.Menu menu1;
java.awt.MenuItem miNew;
java.awt.MenuItem miExit;
java.awt.Menu menu2;
java.awt.MenuItem miAbout;
//}}
```

```
//-------------------------------------------------------------------------------
//Class:      SymMouse
//Purpose:    Inherits methods from the Java class MouseAdapter.  This class is an event
//            listener which should be attached to GUI components that are part of the applet.
//            It overrides the method mouseClicked, checks for the source of the event,
//            and dispatches to an appropriate procedure based on the event's source.  Listeners
//            should be registered at the time the component is put onto the applet, i.e. in the
//            applet init() method.
//Exceptions: None
//-------------------------------------------------------------------------------
class SymMouse extends java.awt.event.MouseAdapter
    {
            public void mouseClicked(java.awt.event.MouseEvent event)
            {
              Object object = event.getSource();
                if (object == submitButton)
                   submitButton_MouseClick(event);
            }
    }
```

```
//-------------------------------------------------------------------------------
//Function:    submitButton_MouseClick()
//Parameters:  event - the MouseEvent that occurred
//Return Type: None
//Purpose:     Checks to see if the applet is processing information.  If it is, no actions is
//             taken.  If not, then the client's data is checked.
```

239

```java
//Exceptions:  None
//----------------------------------------------------------------------------------
        public void submitButton_MouseClick(java.awt.event.MouseEvent event)
        {
    if (!processing){
      processing = true;
      responseText.appendText("Validating data...\n");
      checkData();
    }
        }


//----------------------------------------------------------------------------------
//Function:       checkData()
//Parameters:  None
//Return Type: None
//Purpose:     Checks the client's data and alerts the client as to what errors were found.
//Exceptions:  None
//----------------------------------------------------------------------------------
private void checkData(){

  java.lang.String flight_hrs = flightHoursText.getText();
  java.lang.String type_hrs = typeHoursText.getText();
  int error_count = 0;
  boolean any_errors = false;
  boolean number_format_error = false;

  //clear the response box
  responseText.setText("");

  //check bio page
  responseText.appendText("CHECKING BIO:\n");

  if (rankChoice.getSelectedIndex() == -1){
    responseText.appendText(Integer.toString(++error_count) + ". No rank selected.\n");
    any_errors = true;
  }

  if (designationChoice.getSelectedIndex() == -1){
    responseText.appendText(Integer.toString(++error_count) + ". Missing designator.\n");
    any_errors = true;
  }

  if (communityChoice.getSelectedIndex() == -1){
    responseText.appendText(Integer.toString(++error_count) + ". No community selected.\n");
    any_errors = true;
  }

  if (flight_hrs.compareTo("") == 0){
    responseText.appendText(Integer.toString(++error_count) + ". Flight hours was not entered!\n");
    any_errors = true;
  }

  if (type_hrs.compareTo("") == 0){
    responseText.appendText(Integer.toString(++error_count) + ". Type hours was not entered!\n");
    any_errors = true;
  }
```

```java
try {
  if ((flight_hrs.compareTo("") != 0) &&
              ((Integer.parseInt(flight_hrs) <= 0) || (Integer.parseInt(flight_hrs) >= 10000))){
    responseText.appendText(Integer.toString(++error_count) + ". Flight hours must be between 0
    and 10000!\n");
    any_errors = true;
  }
} catch (java.lang.NumberFormatException e){
    responseText.appendText(Integer.toString(++error_count) + ". Flight hours is not a number!
    (Ensure you round to the nearest hour)\n");
  number_format_error = any_errors = true;
}

try {
  if ((type_hrs.compareTo("") != 0) &&
    ((Integer.parseInt(type_hrs) <= 0) || (Integer.parseInt(type_hrs) >= 10000))){
    responseText.appendText(Integer.toString(++error_count) + ". Type hours must be between 0 and
    10000!\n");
    any_errors = true;
  }
} catch (java.lang.NumberFormatException e){
    responseText.appendText(Integer.toString(++error_count) + ". Type hours is not a number!
    (Ensure you round to the nearest hour)\n");
  number_format_error = any_errors = true;
}

if ((!any_errors) && (!number_format_error)){
  if (Integer.parseInt(flight_hrs) < Integer.parseInt(type_hrs)){
    responseText.appendText(Integer.toString(++error_count) +
    ". Flight hours must be greater than type hours!\n");
    any_errors = true;
  }
}

if (DepartmentHead.getSelectedCheckbox() == null){
  responseText.appendText(Integer.toString(++error_count) +
    ". You did not indicate whether you are a department head or not.\n");
  any_errors = true;
}

if (status.getSelectedCheckbox() == null){
  responseText.appendText(Integer.toString(++error_count) + ". Status not indicated.\n");
  any_errors = true;
}

if (serviceChoice.getSelectedIndex() == -1){
  responseText.appendText(Integer.toString(++error_count) + ". No service selected.\n");
  any_errors = true;
}

if (location.getSelectedCheckbox() == null){
  responseText.appendText(Integer.toString(++error_count) +
    ". You must indicate your unit's location.\n");
  any_errors = true;
}
```

241

```java
    if (!any_errors)
        responseText.appendText("Bio questions answered correctly.");

    responseText.appendText("\nCHECKING QUESTIONS:\n");

    boolean unanswered_questions = false;

    //check all questions
    for (int ix=0; ix <NUM_QUESTIONS; ix++){
        if (choices[ix].getSelectedIndex() == -1)
            any_errors = unanswered_questions = true;
    }

    if(unanswered_questions){
        responseText.appendText
            ("PLEASE ENTER ANSWERS FOR THE FOLLOWING QUESTIONS: ");
        for (int jx=0; jx <NUM_QUESTIONS; jx++){
            if (choices[jx].getSelectedIndex() == -1)
                responseText.appendText(Integer.toString(jx+1) + " ");
        }
    }
    else
        responseText.appendText("All questions answered.\n");

    if (!any_errors){
        UserInput ui = new UserInput(this, true);
        ui.setVisible(true);
        java.lang.String squadNum = ui.squadronNumber.getText();
        java.lang.String surveyNum = ui.surveyNumber.getText();
        java.lang.String unit = new String();
        responseText.appendText("No errors! Attempting to submit...");
        boolean valid = checkSurveyNumber(unit, squadNum, surveyNum);
        if (valid)
            makeDatabaseEntry(squadNum, surveyNum);
    }

    processing = false;
}//end check_Data()

//--------------------------------------------------------------------------------
//Function:    checkSurveyNumber()
//Parameters:  unit - the unit submitting the survey
//             squad - the unit's squadron number
//             surv - the individual's survey number
//Return Type: boolean - true if the survey number is valid for the given unit and squadron,
//                       false otherwise
//Purpose:     Confirms if the survey number is valid or not.  If it is valid, then the
//             client can submit the survey, otherwise, the client is informed that the
//             survey number is invalid.
//--------------------------------------------------------------------------------
private boolean checkSurveyNumber
    (java.lang.String unit, java.lang.String squad, java.lang.String surv){

    java.lang.String communities[] = {"HC", "HCS", "HM", "HMH", "HMT", "HS", "HSC", "HSL",
                    "VAQ", "VAW", "VF", "VFA", "VMAQ", "VMFA",
```

242

```
                              "VP", "VQ", "VR", "VRC", "VS", "VT",
                              "VX", "Other"};

    java.lang.String url = "jdbc:ff-microsoft://spitfire.avsafety.nps.navy.mil:1433/Num_List_DB";
    int start = -1, stop = -1;

    boolean valid = false;

    if (communityChoice.getSelectedIndex() != -1){
      unit = new String(communities[communityChoice.getSelectedIndex()] + squad);

      try {
        Class.forName("connect.microsoft.MicrosoftDriver").newInstance();
        java.sql.Connection con = DriverManager.getConnection(url, "sa", "");
        java.sql.PreparedStatement stmt = con.prepareStatement
                ("SELECT * FROM Num_List WHERE Unit_ID = ?");
        stmt.setString(1, unit);
        java.sql.ResultSet rs = stmt.executeQuery();

        if (rs != null){
          while (rs.next()){
            rs.getString(1);
            start = rs.getInt(2);
            stop = rs.getInt(3);
          }
        }

        if ((Integer.parseInt(surv) >= start) && (Integer.parseInt(surv) <=stop)){
          valid = true;
          responseText.appendText("Your survey number is valid... Please wait...\n");
        }
        else responseText.appendText("You entered an INVALID number!!!\n");

        stmt.close();
        con.close();
      }
      catch (Exception e) {
        responseText.appendText(e.toString() + "\n");
      }
    }

    return valid;
}//end checkSurveyNumber()


//---------------------------------------------------------------------------------
//Function:   makeDatabaseEntry()
//Parameters: squadN - the unit's squadron number
//            surveyN - the unit's survey number
//Return Type: None
//Purpose:    The client's data has already been checked, and the survey number has been
//            been validated prior to this function being called.  Once this procedure is
//            completed, the client's data has been inserted into the SQL 6.5 database.
//            Entries are made in both the biographical and questions table, and if either
//            insertion operation fails, then the transaction is rolled back (i.e. it is
//            removed from the database).
```

```java
//---------------------------------------------------------------------------
private void makeDatabaseEntry(java.lang.String squadN, java.lang.String surveyN){

    java.lang.String url = "jdbc:ff-microsoft://spitfire.avsafety.nps.navy.mil:1433/Survey2_DB";

    java.lang.String designator[] = {"Pilot", "NFO", "Other"};
    java.lang.String communities[] = {"HC", "HCS", "HM", "HMH", "HMT", "HS", "HSC", "HSL",
                    "VAQ", "VAW", "VF", "VFA", "VMAQ", "VMFA",
                    "VP", "VQ", "VR", "VRC", "VS", "VT",
                    "VX", "Other"};

    java.lang.String service[] = {"USN", "USMC", "Other"};

    java.lang.String DH, ST, LOC;

    //independently determine which checkboxes are selected
    if (DepartmentHead.getSelectedCheckbox() == yesRadioButton)
      DH = new String("Yes");
    else DH = new String("No");

    if (status.getSelectedCheckbox() == regularRadioButton)
      ST = new String("Regular");
    else if (status.getSelectedCheckbox() == activeRadioButton)
      ST = new String("Active Reserve");
    else ST = new String("Drilling Reserve");

    if (location.getSelectedCheckbox() == eastButton)
      LOC = new String("East Coast");
    else if (location.getSelectedCheckbox() == westButton)
      LOC = new String("West Coast");
    else LOC = new String("Other");

    java.sql.Connection con = null;

    try {
      Class.forName("connect.microsoft.MicrosoftDriver").newInstance();
      Driver driver2 = new connect.microsoft.MicrosoftDriver();
      con = DriverManager.getConnection(url, "sa", "");
      con.setAutoCommit(false);
      java.sql.Statement stmt1 = con.createStatement();
      java.sql.Statement stmt2 = con.createStatement();

      stmt1.executeUpdate("INSERT INTO Biographical_Data VALUES('" +
                  communities[communityChoice.getSelectedIndex()] + "'," +
                  Integer.parseInt(squadN) + "," +
                  Integer.parseInt(surveyN) +"," +
                  rankChoice.getSelectedIndex() + ",'" +
                  designator[designationChoice.getSelectedIndex()] +"'," +
                  Integer.parseInt(flightHoursText.getText()) +"," +
                  Integer.parseInt(typeHoursText.getText()) + ",'" +
                  DH + "','" +
                  ST + "','" +
                  service[serviceChoice.getSelectedIndex()] + "','" +
                  LOC + "')");

      java.lang.String operation = "INSERT INTO Questions VALUES('" +
```

```java
            communities[communityChoice.getSelectedIndex()] + "'," +
            Integer.parseInt(squadN) + "," +
            Integer.parseInt(surveyN) +",";
    for (int ix=0; ix<NUM_QUESTIONS-1; ix++)
        operation += (choices[ix].getSelectedIndex() + ",");

    operation += choices[NUM_QUESTIONS-1].getSelectedIndex() + ")";

    stmt2.executeUpdate(operation);
    con.commit();
    con.setAutoCommit(true);
    stmt1.close();
    stmt2.close();
    con.close();
    responseText.appendText("Survey successfully submitted. Thank you for your input.\n");
    this.hide();
    parent.getAppletContext().showDocument
        (new URL("http://spitfire.avsafety.nps.navy.mil/ThankYou.html"), "_self");
    }
    catch (Exception e){
    responseText.appendText("Survey not submitted!  Processing error encountered:\n");
    responseText.appendText(e.toString() + "\n");
    try {
        if (con != null)
            con.rollback();
    }
    catch (SQLException ex){
        responseText.appendText("Survey not submitted!  Processing error encountered:\n");
        responseText.appendText(ex.toString() + "\n");
    }
    }
}//end makeDatabaseEntry()



class SymWindow extends java.awt.event.WindowAdapter
{
    public void windowClosing(java.awt.event.WindowEvent event)
        {
            Object object = event.getSource();
            if (object == Questionnaire.this)
                    Questionnaire_WindowClosing(event);
        }
}//end class SymWindow



void Questionnaire_WindowClosing(java.awt.event.WindowEvent event)
{
    hide();          // hide the Frame
        dispose();           // free the system resources
        System.exit(0);  // close the application
}//end Questionnaire_WindowClosing()


class SymAction implements java.awt.event.ActionListener
{
    public void actionPerformed(java.awt.event.ActionEvent event)
```

```java
        {
          Object object = event.getSource();
          if (object == miAbout)
                miAbout_Action(event);
            else if (object == miExit)
                miExit_Action(event);
        else if (object == miNew)
                miNew_ActionPerformed(event);
        }
    }//end class SymAction


    void miAbout_Action(java.awt.event.ActionEvent event)
    {
      //{{CONNECTION
          // Action from About Create and show as modal
          AboutDialog ad = new AboutDialog(this, true);
        ad.setVisible(true);
          //}}
    }//end miAbout_Action()


    void miExit_Action(java.awt.event.ActionEvent event)
    {
      //{{CONNECTION
          // Action from Exit Create and show as modal
      (new QuitDialog(this, true)).setVisible(true);
          //}}
    }//end miExitAction()


    void miNew_ActionPerformed(java.awt.event.ActionEvent event)
    {
      try {
        rankChoice.select(-1);
        designationChoice.select(-1);
        communityChoice.select(-1);
        flightHoursText.setText("");
        typeHoursText.setText("");
        DepartmentHead.setCurrent(null);
        status.setCurrent(null);
        serviceChoice.select(-1);
        location.setCurrent(null);
        for (int ix=0; ix<NUM_QUESTIONS; ix++)
          choices[ix].select(-1);
      }
      catch (IllegalArgumentException e) { }
    }//end miNew_ActionPerformed()

}//end class Questionnaire



C.    USERINPUT.JAVA SOURCE CODE
import java.awt.*;

public class UserInput extends Dialog
```

```
{
    public UserInput(Frame parent, boolean modal)
    {
        super(parent, modal);

        // This code is automatically generated by Visual Cafe when you add
        // components to the visual environment. It instantiates and initializes
        // the components. To modify the code, only use code syntax that matches
        // what Visual Cafe can generate, or Visual Cafe may be unable to back
        // parse your Java file into its visual environment.
        //{{INIT_CONTROLS
        setLayout(null);
        setVisible(false);
        setSize(430,257);
        setResizable(false);
        setBackground(new Color(12632256));
        label1 = new java.awt.Label("Access Number:",Label.RIGHT);
        label1.setBounds(24,36,168,54);
        label1.setFont(new Font("Dialog", Font.PLAIN, 18));
        add(label1);
        label2 = new java.awt.Label("Survey Number:",Label.RIGHT);
        label2.setBounds(24,96,168,54);
        label2.setFont(new Font("Dialog", Font.PLAIN, 18));
        add(label2);
        squadronNumber = new java.awt.TextField();
        squadronNumber.setBounds(204,48,186,36);
        squadronNumber.setFont(new Font("Dialog", Font.PLAIN, 22));
        add(squadronNumber);
        surveyNumber = new java.awt.TextField();
        surveyNumber.setBounds(204,108,186,36);
        surveyNumber.setFont(new Font("Dialog", Font.PLAIN, 22));
        add(surveyNumber);
        OKButton = new java.awt.Button();
        OKButton.setLabel("OK");
        OKButton.setBounds(180,168,97,48);
        OKButton.setFont(new Font("Dialog", Font.PLAIN, 22));
        OKButton.setForeground(new Color(255));
        OKButton.setBackground(new Color(12632256));
        add(OKButton);
        setTitle("The following information is requested:");
        //}}

        //{{REGISTER_LISTENERS
        SymWindow aSymWindow = new SymWindow();
        this.addWindowListener(aSymWindow);
        SymMouse aSymMouse = new SymMouse();
        OKButton.addMouseListener(aSymMouse);
        //}}
    }

    public void addNotify()
    {
        // Record the size of the window prior to calling parents addNotify.
        Dimension d = getSize();

        super.addNotify();
```

247

```java
        if (fComponentsAdjusted)
                return;

        // Adjust components according to the insets
        setSize(insets().left + insets().right + d.width, insets().top + insets().bottom + d.height);
        Component components[] = getComponents();
        for (int i = 0; i < components.length; i++)
        {
                Point p = components[i].getLocation();
                p.translate(insets().left, insets().top);
                components[i].setLocation(p);
        }
        fComponentsAdjusted = true;
    }

// Used for addNotify check.
    boolean fComponentsAdjusted = false;


    public UserInput(Frame parent, String title, boolean modal)
    {
            this(parent, modal);
            setTitle(title);
    }

/**
 * Shows or hides the component depending on the boolean flag b.
 * @param b  if true, show the component; otherwise, hide the component.
 * @see java.awt.Component#isVisible
 */
public void setVisible(boolean b)
    {
            if(b)
            {
                    Rectangle bounds = getParent().getBounds();
                    Rectangle abounds = getBounds();

                    setLocation(bounds.x + (bounds.width - abounds.width)/ 2,
                            bounds.y + (bounds.height - abounds.height)/2);
            }
            super.setVisible(b);
    }

    //{{DECLARE_CONTROLS
    java.awt.Label label1;
    java.awt.Label label2;
    java.awt.TextField squadronNumber;
    java.awt.TextField surveyNumber;
    java.awt.Button OKButton;
    //}}

    class SymWindow extends java.awt.event.WindowAdapter
    {
            public void windowClosing(java.awt.event.WindowEvent event)
            {
```

```java
                        Object object = event.getSource();
                        if (object == UserInput.this)
                                UserInput_WindowClosing(event);
                }
        }

        void UserInput_WindowClosing(java.awt.event.WindowEvent event)
        {
                //setVisible(false);
        }

        class SymMouse extends java.awt.event.MouseAdapter
        {
                public void mouseClicked(java.awt.event.MouseEvent event)
                {
                        Object object = event.getSource();
                        if (object == OKButton)
                                OKButton_MouseClicked(event);
                }
        }

        void OKButton_MouseClicked(java.awt.event.MouseEvent event) throws
                java.lang.NumberFormatException
        {
          java.lang.String surveyN = surveyNumber.getText();
          java.lang.String squadN = squadronNumber.getText();
          try {
            if ((surveyN.compareTo("") == 0) || (squadN.compareTo("") == 0) ||
              (Integer.parseInt(surveyN) < 0) || (Integer.parseInt(squadN) < 0)){
            squadronNumber.requestFocus();
            squadronNumber.setText("");
            surveyNumber.setText("");
            }
        else dispose();
          }
          catch (java.lang.NumberFormatException e){
        squadronNumber.requestFocus();
        squadronNumber.setText("");
        surveyNumber.setText("");
          }
        }
}
```

D.   ABOUTDIALOG.JAVA SOURCE CODE

```java
import java.awt.*;

public class AboutDialog extends Dialog {

        public AboutDialog(Frame parent, boolean modal)
        {
                super(parent, modal);

                // This code is automatically generated by Visual Cafe when you add
                // components to the visual environment. It instantiates and initializes
                // the components. To modify the code, only use code syntax that matches
                // what Visual Cafe can generate, or Visual Cafe may be unable to back
```

```java
// parse your Java file into its visual environment.

//{{INIT_CONTROLS
setLayout(null);
setVisible(false);
setSize(377,233);
setBackground(new Color(255));
label1 = new java.awt.Label("School of Aviation Safety",Label.CENTER);
label1.setBounds(60,24,248,21);
label1.setFont(new Font("Dialog", Font.BOLD, 14));
label1.setForeground(new Color(16776960));
add(label1);
okButton = new java.awt.Button();
okButton.setLabel("OK");
okButton.setBounds(156,156,66,27);
okButton.setBackground(new Color(12632256));
add(okButton);
label2 = new java.awt.Label("Naval Post Graduate School",Label.CENTER);
label2.setBounds(60,48,248,21);
label2.setFont(new Font("Dialog", Font.BOLD, 14));
label2.setForeground(new Color(16776960));
add(label2);
label3 = new java.awt.Label("Monterey, California",Label.CENTER);
label3.setBounds(60,72,248,21);
label3.setFont(new Font("Dialog", Font.BOLD, 14));
label3.setForeground(new Color(16776960));
add(label3);
label4 = new java.awt.Label("Phone: 831-656-2581",Label.CENTER);
label4.setBounds(60,96,248,21);
label4.setFont(new Font("Dialog", Font.BOLD, 14));
label4.setForeground(new Color(16776960));
add(label4);
label5 = new java.awt.Label("E-mail: aciavarelli@nps.navy.mil",Label.CENTER);
label5.setBounds(57,120,248,21);
label5.setFont(new Font("Dialog", Font.BOLD, 14));
label5.setForeground(new Color(16776960));
add(label5);
setTitle("About");
setResizable(false);
//}}

//{{REGISTER_LISTENERS
SymWindow aSymWindow = new SymWindow();
this.addWindowListener(aSymWindow);
SymAction lSymAction = new SymAction();
okButton.addActionListener(lSymAction);
//}}

}

public AboutDialog(Frame parent, String title, boolean modal)
{
        this(parent, modal);
        setTitle(title);
}
```

```java
public void addNotify()
{
        // Record the size of the window prior to calling parents addNotify.
    Dimension d = getSize();

        super.addNotify();

        // Only do this once.
        if (fComponentsAdjusted)
                return;

        // Adjust components according to the insets
        setSize(insets().left + insets().right + d.width, insets().top + insets().bottom + d.height);
        Component components[] = getComponents();
        for (int i = 0; i < components.length; i++)
        {
                Point p = components[i].getLocation();
                p.translate(insets().left, insets().top);
                components[i].setLocation(p);
        }

        // Used for addNotify check.
        fComponentsAdjusted = true;
}

public void setVisible(boolean b)
{
    if (b)
    {
        Rectangle bounds = getParent().bounds();
        Rectangle abounds = bounds();

        move(bounds.x + (bounds.width - abounds.width)/ 2,
                bounds.y + (bounds.height - abounds.height)/2);
    }

        super.setVisible(b);
}

//{{DECLARE_CONTROLS
java.awt.Label label1;
java.awt.Button okButton;
java.awt.Label label2;
java.awt.Label label3;
java.awt.Label label4;
java.awt.Label label5;
//}}

// Used for addNotify check.
    boolean fComponentsAdjusted = false;

    class SymWindow extends java.awt.event.WindowAdapter
    {
            public void windowClosing(java.awt.event.WindowEvent event)
            {
                    Object object = event.getSource();
```

```
                if (object == AboutDialog.this)
                        AboutDialog_WindowClosing(event);
        }
}

void AboutDialog_WindowClosing(java.awt.event.WindowEvent event)
{
 dispose();
}

class SymAction implements java.awt.event.ActionListener
{
        public void actionPerformed(java.awt.event.ActionEvent event)
        {
                Object object = event.getSource();
                if (object == okButton)
                        okButton_Clicked(event);
        }
}

void okButton_Clicked(java.awt.event.ActionEvent event)
{
        //{{CONNECTION
        // Clicked from okButton Hide the Dialog
 dispose();
        //}}
        }
}
```

E.   QUITDIALOG.JAVA SOURCE CODE

```
import java.awt.*;
import java.awt.event.*;

public class QuitDialog extends Dialog
{
        public QuitDialog(Frame parent, boolean modal)
        {
                super(parent, modal);
                //{{INIT_CONTROLS
                setLayout(null);
                setVisible(false);
                setSize(337,135);
                setBackground(new Color(16711680));
                yesButton = new java.awt.Button();
                yesButton.setLabel(" Yes ");
                yesButton.setBounds(72,80,79,22);
                yesButton.setFont(new Font("Dialog", Font.BOLD, 12));
                yesButton.setBackground(new Color(12632256));
                add(yesButton);
                noButton = new java.awt.Button();
                noButton.setLabel(" No ");
                noButton.setBounds(185,80,79,22);
                noButton.setFont(new Font("Dialog", Font.BOLD, 12));
                noButton.setBackground(new Color(12632256));
                add(noButton);
                label1 = new java.awt.Label("Do you really want to quit?",Label.CENTER);
```

252

```java
        label1.setBounds(48,36,246,23);
        label1.setFont(new Font("Dialog", Font.BOLD, 16));
        label1.setForeground(new Color(16776960));
        add(label1);
        setTitle("ACSA Survey - Quit");
        setResizable(false);
        fparent = parent;
        //}}

        //{{REGISTER_LISTENERS
        SymWindow aSymWindow = new SymWindow();
        this.addWindowListener(aSymWindow);
        SymAction lSymAction = new SymAction();
        noButton.addActionListener(lSymAction);
        yesButton.addActionListener(lSymAction);
        //}}
    }

    public void addNotify()
    {
        // Record the size of the window prior to calling parents addNotify.
        Dimension d = getSize();

        super.addNotify();

        if (fComponentsAdjusted)
            return;

        // Adjust components according to the insets
        setSize(insets().left + insets().right + d.width, insets().top + insets().bottom + d.height);
        Component components[] = getComponents();
        for (int i = 0; i < components.length; i++)
        {
            Point p = components[i].getLocation();
            p.translate(insets().left, insets().top);
            components[i].setLocation(p);
        }
        fComponentsAdjusted = true;
    }

    public QuitDialog(Frame parent, String title, boolean modal)
    {
        this(parent, modal);
        setTitle(title);
    }

/**
 * Shows or hides the component depending on the boolean flag b.
 * @param b  if true, show the component; otherwise, hide the component.
 * @see java.awt.Component#isVisible
 */
public void setVisible(boolean b)
    {
        if(b)
        {
            Rectangle bounds = getParent().getBounds();
```

```java
                    Rectangle abounds = getBounds();

                    setLocation(bounds.x + (bounds.width - abounds.width)/ 2,
                            bounds.y + (bounds.height - abounds.height)/2);
            }
            super.setVisible(b);
    }

// Used for addNotify check.
      boolean fComponentsAdjusted = false;

      //{{DECLARE_CONTROLS
      java.awt.Button yesButton;
      java.awt.Button noButton;
      java.awt.Label label1;
      java.awt.Frame fparent;
      //}}

      class SymWindow extends java.awt.event.WindowAdapter
      {
              public void windowClosing(java.awt.event.WindowEvent event)
              {
                      Object object = event.getSource();
                      if (object == QuitDialog.this)
                              QuitDialog_WindowClosing(event);
              }
      }

      void QuitDialog_WindowClosing(java.awt.event.WindowEvent event)
      {
      dispose();
      }

      class SymAction implements java.awt.event.ActionListener
      {
              public void actionPerformed(java.awt.event.ActionEvent event)
              {
                      Object object = event.getSource();
                      if (object == noButton)
                              noButton_Clicked(event);
                      else if (object == yesButton)
                              yesButton_Clicked(event);
              }
      }

      void yesButton_Clicked(java.awt.event.ActionEvent event)
      {
              dispose();
        fparent.dispose();
      }

      void noButton_Clicked(java.awt.event.ActionEvent event)
      {
        dispose();
      }
}
```

APPENDIX H. ACTIVE SERVER PAGES (ASP) SOURCE CODE

Because most of the ASP code was self-generated by Visual Interdev, this appendix only contains the ASP portion that was developed to work in conjunction with the Secure Socket Layer (SSL) protocol. The remaining ASP pages can be found in the *D:\Intetpub\wwwroot\admin* subdirectory of the *spitfire* machine. These files either begin with the prefix *SurveyNumbers*, which are used with the *Num_List_DB*, or by *Survey*, which are used with the *Survey2_DB*.

```
<%@ LANGUAGE="VBSCRIPT" %>
<!--#include file="ADOVBS.INC" -->
<%
  Survey2_DB_ConnectionString = "DSN=Survey2_DB;Description=Survey
Database;SERVER=SPITFIRE;UID=sa;PWD=;APP=Microsoft (R) Developer
Studio;WSID=SPITFIRE;DATABASE=Survey2_DB"
  DB_ConnectionTimeout = 15
  DB_CommandTimeout = 30
  DB_RuntimeUserName = "sa"
  DB_RuntimePassword = ""

  Num_List_DB_ConnectionString = "DSN=Num_List;Description=squadron survey access
listing;SERVER=SPITFIRE;UID=sa;PWD=;APP=Microsoft (R) Developer
Studio;WSID=SPITFIRE;DATABASE=Num_List_DB"

  Set connection = Server.CreateObject("ADODB.Connection")
  connection.ConnectionTimeout = DB_ConnectionTimeout
  connection.CommandTimeout = DB_CommandTimeout
  connection.Open Num_List_DB_ConnectionString, DB_RuntimeUserName, DB_RuntimePassword

  community = Request.Form("community")
  squadron_number = Request.Form("squadron_number")
  survey_number = Request.Form("survey_number")
  rank = Request.Form("rank")
  designator = Request.Form("designator")
  flight_hours = Request.Form("flight_hours")
  type_hours = Request.Form("type_hours")
  dept_head = Request.Form("dept_head")
  status = Request.Form("status")
  service = Request.Form("service")
  location = Request.Form("location")

  concatString = community & squadron_number
  valid_survey = false
  start_number = -1
  stop_number = -1

  query = "SELECT * FROM NUM_LIST WHERE Unit_ID = '" & concatString & "'"
  Set records = Server.CreateObject("ADODB.RecordSet")
  Set records = connection.Execute (query, RecordsAffected, adCmdText)
```

```
if (NOT (records.EOF)) then
   for each field in records.Fields
      if field.Name = "Start_Number" then
         start_number = CInt(field.Value)
      end if
      if field.Name = "End_Number" then
         stop_number = CInt(field.Value)
      end if
   next
   if (CInt(survey_number) >= start_number) AND (CInt(survey_number) <= stop_number) then
      valid_survey = true
   else
      Response.Redirect "https://spitfire.avsafety.nps.navy.mil/NumberError.html"
   end if
end if


connection.Close


if (valid_survey) then
   connection.Open Survey2_DB_ConnectionString, DB_RuntimeUserName, DB_RuntimePassword
   biodata = "INSERT INTO Biographical_Data VALUES ('" & community & "'," &_
         squadron_number & "," & survey_number & "," & rank & ",'" & designator & "'," &_
         flight_hours & "," & type_hours & ",'" & dept_head & "','" & status & "','" & service & "','" &_
         location & "')"

   question1 = Request.Form("question1")
   question2 = Request.Form("question2")
   question3 = Request.Form("question3")
   question4 = Request.Form("question4")
   question5 = Request.Form("question5")
   question6 = Request.Form("question6")
   question7 = Request.Form("question7")
   question8 = Request.Form("question8")
   question9 = Request.Form("question9")
   question10 = Request.Form("question10")
   question11 = Request.Form("question11")
   question12 = Request.Form("question12")
   question13 = Request.Form("question13")
   question14 = Request.Form("question14")
   question15 = Request.Form("question15")
   question16 = Request.Form("question16")
   question17 = Request.Form("question17")
   question18 = Request.Form("question18")
   question19 = Request.Form("question19")
   question20 = Request.Form("question20")
   question21 = Request.Form("question21")
   question22 = Request.Form("question22")
   question23 = Request.Form("question23")
   question24 = Request.Form("question24")
   question25 = Request.Form("question25")
   question26 = Request.Form("question26")
   question27 = Request.Form("question27")
   question28 = Request.Form("question28")
   question29 = Request.Form("question29")
   question30 = Request.Form("question30")
   question31 = Request.Form("question31")
```

```
question32 = Request.Form("question32")
question33 = Request.Form("question33")
question34 = Request.Form("question34")
question35 = Request.Form("question35")
question36 = Request.Form("question36")
question37 = Request.Form("question37")
question38 = Request.Form("question38")
question39 = Request.Form("question39")
question40 = Request.Form("question40")
question41 = Request.Form("question41")
question42 = Request.Form("question42")
question43 = Request.Form("question43")
question44 = Request.Form("question44")
question45 = Request.Form("question45")
question46 = Request.Form("question46")
question47 = Request.Form("question47")
question48 = Request.Form("question48")
question49 = Request.Form("question49")
question50 = Request.Form("question50")
question51 = Request.Form("question51")
question52 = Request.Form("question52")
question53 = Request.Form("question53")
question54 = Request.Form("question54")
question55 = Request.Form("question55")
question56 = Request.Form("question56")
question57 = Request.Form("question57")
question58 = Request.Form("question58")
question59 = Request.Form("question59")
question60 = Request.Form("question60")
question61 = Request.Form("question61")

questiondata = "INSERT INTO Questions VALUES ('" & community & "'," &
        squadron_number & "," & survey_number & "," &_
        question1 & "," & question2 & "," & question3 & "," & question4 & "," &_
        question5 & "," & question6 & "," & question7 & "," & question8 & "," & question9 & "," &_
        question10 & "," & question11 & "," & question12 & "," & question13 & "," &_
        question14 & "," & question15 & "," & question16 & "," & question17 & "," &_
        question18 & "," & question19 & "," & question20 & "," & question21 & "," &_
        question22 & "," & question23 & "," & question24 & "," & question25 & "," &_
        question26 & "," & question27 & "," & question28 & "," & question29 & "," &_
        question30 & "," & question31 & "," & question32 & "," & question33 & "," &_
        question34 & "," & question35 & "," & question36 & "," & question37 & "," &_
        question38 & "," & question39 & "," & question40 & "," & question41 & "," &_
        question42 & "," & question43 & "," & question44 & "," & question45 & "," &_
        question46 & "," & question47 & "," & question48 & "," & question49 & "," &_
        question50 & "," & question51 & "," & question52 & "," & question53 & "," &_
        question54 & "," & question55 & "," & question56 & "," & question57 & "," &_
        question58 & "," & question59 & "," & question60 & "," & question61 & ")"

connection.BeginTrans
connection.Execute biodata, RecordsAffected, adCmdText
connection.Execute questiondata, RecordsAffected, adCmdText
If connection.Errors.Count = 0 then
  connection.CommitTrans
  connection.Close
  Response.Redirect "https://spitfire.avsafety.nps.navy.mil/ThankYou.html"
```

```
    else
      connection.RollbackTrans
      connection.Close
    end if

  else
    Response.Redirect "https://spitfire.avsafety.nps.navy.mil/NumberError.html"
  end if
%>
<HTML><HEAD><TITLE>Processing survey data</TITLE></HEAD>
<BODY>
</BODY>
</HTML>
```

APPENDIX I. ACSA PROJECT DIRECTORY STRUCTURE

Directory of D:\Inetpub\wwwroot

```
[.]                [..]               [admin]            [cgi-bin]
[classes]          default.asp        file               file2
flup1-alpha.cab    flup1-x86.cab      flup1.cab          Fortune.html
[Help]             [images]           index.asp          [manual]
NumberError.html   POSTINFO.EG        postinfo.html      [Results]
[secureSurvey]     [security]         [servletrunner]    [SERVLETS]
[sounds]           [Survey]           [surveyadmin]      ThankYou.html
[time]             wpie15-alpha.cab   wpie15-x86.cab     [_private]
              32 File(s)        2,133,903 bytes
```

Directory of D:\Inetpub\wwwroot\admin

```
[.]                      [..]                   default.html
global.asa               [images]               SurveyAction.asp
SurveyForm.asp           SurveyList.asp         SurveyNumbersAction.asp
SurveyNumbersForm.asp    SurveyNumbersList.asp
              11 File(s)          163,713 bytes
```

Directory of D:\Inetpub\wwwroot\admin\images

```
[.]                [..]               administrator.gif   Back3.gif
line.gif           NAV1.JPG           point.gif           smblue_rock.gif
vertical_line.gif
              9 File(s)          21,692 bytes
```

Directory of D:\Inetpub\wwwroot\admin_vti_bin

```
fpcount.exe    shtml.dll
              2 File(s)          121,616 bytes
```

Directory of D:\Inetpub\wwwroot\admin_vti_bin_vti_adm

```
admin.dll
              1 File(s)          15,120 bytes
```

Directory of D:\Inetpub\wwwroot\admin_vti_bin_vti_aut

```
author.dll    dvwssr.dll
              2 File(s)          21,536 bytes
```

Directory of D:\Inetpub\wwwroot\admin_vti_cnf

```
global.asa               SurveyNumbersAction.asp    SurveyNumbersForm.asp
SurveyNumbersList.asp
              4 File(s)          3,354 bytes
```

Directory of D:\Inetpub\wwwroot\admin_vti_pvt

```
access.cnf    deptodoc.btr    doctodep.btr    linkinfo.cnf    service.cnf
service.lck
              6 File(s)          7,461 bytes
```

```
Directory of D:\Inetpub\wwwroot\cgi-bin

[.]              [..]              Count.exe       extdgts.exe     extdgts.txt
htimage.exe      imagemap.exe      mkstrip.exe     mkstrip.txt     [wcount]
            10 File(s)              277,438 bytes

Directory of D:\Inetpub\wwwroot\cgi-bin\wcount

[.]         [..]         [conf]     [data]     [digits]  [logs]    rgb.txt
            7 File(s)              16,992 bytes

Directory of D:\Inetpub\wwwroot\cgi-bin\wcount\conf

[.]              [..]              count.cfg
            3 File(s)               1,521 bytes

Directory of D:\Inetpub\wwwroot\cgi-bin\wcount\data

[.]              [..]              count.dat      sample.dat
            4 File(s)                30 bytes

Directory of D:\Inetpub\wwwroot\cgi-bin\wcount\digits

[.]  [..] [A]  [B]  [C]  [D]  [E]
            7 File(s)                 0 bytes

Directory of D:\Inetpub\wwwroot\cgi-bin\wcount\digits\A

[.]              [..]              strip.gif   [_vti_cnf]
            4 File(s)                494 bytes

Directory of D:\Inetpub\wwwroot\cgi-bin\wcount\digits\A\_vti_cnf

[.]              [..]              strip.gif
            3 File(s)                265 bytes

Directory of D:\Inetpub\wwwroot\cgi-bin\wcount\digits\B

[.]              [..]              strip.gif   [_vti_cnf]
            4 File(s)               2,942 bytes

Directory of D:\Inetpub\wwwroot\cgi-bin\wcount\digits\B\_vti_cnf

[.]              [..]              strip.gif
            3 File(s)                266 bytes

Directory of D:\Inetpub\wwwroot\cgi-bin\wcount\digits\C

[.]              [..]              strip.gif   [_vti_cnf]
            4 File(s)               1,178 bytes

Directory of D:\Inetpub\wwwroot\cgi-bin\wcount\digits\C\_vti_cnf

[.]              [..]              strip.gif
            3 File(s)                266 bytes
```

```
Directory of D:\Inetpub\wwwroot\cgi-bin\wcount\digits\D

[.]          [..]          lenna.gif   strip.gif   [_vti_cnf]
             5 File(s)          27,387 bytes

Directory of D:\Inetpub\wwwroot\cgi-bin\wcount\digits\D\_vti_cnf

[.]          [..]          lenna.gif   strip.gif
             4 File(s)          532 bytes

Directory of D:\Inetpub\wwwroot\cgi-bin\wcount\digits\E

[.]          [..]          strip.gif   [_vti_cnf]
             4 File(s)          197 bytes

Directory of D:\Inetpub\wwwroot\cgi-bin\wcount\digits\E\_vti_cnf

[.]          [..]          strip.gif
             3 File(s)          265 bytes

Directory of D:\Inetpub\wwwroot\cgi-bin\wcount\logs

[.]          [..]          Count24.log
             3 File(s)          2,379 bytes

Directory of D:\Inetpub\wwwroot\cgi-bin\_vti_cnf

Count.exe     htimage.exe    imagemap.exe
             3 File(s)          715 bytes

Directory of D:\Inetpub\wwwroot\classes

[.]               [..]               fphover.class     fphoverx.class
fprotate.class    fprotatx.class     LScrollText.class [_vti_cnf]
             8 File(s)          28,265 bytes

Directory of D:\Inetpub\wwwroot\classes\_vti_cnf

[.]               [..]               fphover.class     LScrollText.class
             4 File(s)          216 bytes

Directory of D:\Inetpub\wwwroot\Help

[.]          [..]          Help.html   [images]    [_vti_cnf]
             5 File(s)          15,837 bytes

Directory of D:\Inetpub\wwwroot\Help\images

[.]               [..]               Console.jpg       display.jpg
help.h1.jpg       help.h2.jpg        help.h3.jpg       results.jpg
servleterror1.jpg servleterror2.jpg  [_vti_cnf]
             11 File(s)          229,930 bytes

Directory of D:\Inetpub\wwwroot\Help\images\_vti_cnf

[.]               [..]               Console.jpg       display.jpg
help.h1.jpg       help.h2.jpg        help.h3.jpg       results.jpg
```

```
servleterror1.jpg    servleterror2.jpg
               10 File(s)             2,424 bytes


   Directory of D:\Inetpub\wwwroot\Help\_vti_cnf

[.]            [..]             Help.html
               3 File(s)              796 bytes


   Directory of D:\Inetpub\wwwroot\images

[.]            [..]             BACK1.JPG       bnewb.gif       F18.jpg
flag2.gif      NEW.gif          security.gif    Sh-60.jpg       sumtextb.jpg
telephone.gif  WARNING2.GIF     [_vti_cnf]
               13 File(s)             135,511 bytes


   Directory of D:\Inetpub\wwwroot\images\_vti_cnf

[.]            [..]             BACK1.JPG       F18.jpg         Sh-60.jpg
sumtextb.jpg   telephone.gif    WARNING2.GIF
               8 File(s)              1,767 bytes


   Directory of D:\Inetpub\wwwroot\manual

[.]                            [..]
bind.html                      cgi_path.html
content-negotiation.html       custom-error.html
dns-caveats.html               dso.html
ebcdic.html                    env.html
footer.html                    handler.html
header.html                    [images]
index.html                     install.html
invoking.html                  keepalive.html
LICENSE                        location.html
man-template.html              [misc]
[mod]                          multilogs.html
new_features_1_0.html          new_features_1_1.html
new_features_1_2.html          new_features_1_3.html
process-model.html             sections.html
sourcereorg.html               stopping.html
suexec.html                    unixware.html
upgrading_to_1_3.html          [vhosts]
windows.html                   [_vti_cnf]
               38 File(s)             255,375 bytes


   Directory of D:\Inetpub\wwwroot\manual\images

[.]                  [..]                    custom_errordocs.gif
home.gif             index.gif               mod_rewrite_fig1.fig
mod_rewrite_fig1.gif mod_rewrite_fig2.fig    mod_rewrite_fig2.gif
sub.gif              [_vti_cnf]
               11 File(s)             42,478 bytes


   Directory of D:\Inetpub\wwwroot\manual\images\_vti_cnf

[.]                  [..]                    custom_errordocs.gif
home.gif             index.gif               mod_rewrite_fig1.gif
mod_rewrite_fig2.gif sub.gif
```

```
                8 File(s)           9,427 bytes

  Directory of D:\Inetpub\wwwroot\manual\misc

  [.]                             [..]
  API.html                        client_block_api.html
  compat_notes.html               custom_errordocs.html
  descriptors.html                FAQ.html
  fin_wait_2.html                 footer.html
  header.html                     howto.html
  HTTP_Features.tsv               index.html
  known_client_problems.html      nopgp.html
  perf-bsd44.html                 perf-dec.html
  perf-hp.html                    perf-tuning.html
  perf.html                       security_tips.html
  vif-info.html                   windoz_keepalive.html
  [_vti_cnf]
                25 File(s)          321,774 bytes

  Directory of D:\Inetpub\wwwroot\manual\misc\_vti_cnf

  [.]                             [..]
  API.html                        client_block_api.html
  compat_notes.html               custom_errordocs.html
  descriptors.html                FAQ.html
  fin_wait_2.html                 footer.html
  header.html                     howto.html
  HTTP_Features.tsv               index.html
  known_client_problems.html      nopgp.html
  perf-bsd44.html                 perf-dec.html
  perf-hp.html                    perf-tuning.html
  perf.html                       security_tips.html
  vif-info.html                   windoz_keepalive.html
                24 File(s)           22,566 bytes

  Directory of D:\Inetpub\wwwroot\manual\mod

  [.]                     [..]                    core.html
  directive-dict.html     directives.html         footer.html
  header.html             index.html              mod_access.html
  mod_actions.html        mod_alias.html          mod_asis.html
  mod_auth.html           mod_auth_anon.html      mod_auth_db.html
  mod_auth_dbm.html       mod_autoindex.html      mod_browser.html
  mod_cern_meta.html      mod_cgi.html            mod_cookies.html
  mod_digest.html         mod_dir.html            mod_dld.html
  mod_dll.html            mod_env.html            mod_example.html
  mod_expires.html        mod_headers.html        mod_imap.html
  mod_include.html        mod_info.html           mod_isapi.html
  mod_log_agent.html      mod_log_common.html     mod_log_config.html
  mod_log_referer.html    mod_mime.html           mod_mime_magic.html
  mod_mmap_static.html    mod_negotiation.html    mod_proxy.html
  mod_rewrite.html        mod_setenvif.html       mod_so.html
  mod_speling.html        mod_status.html         mod_unique_id.html
  mod_userdir.html        mod_usertrack.html      [_vti_cnf]
                51 File(s)           523,018 bytes

  Directory of D:\Inetpub\wwwroot\manual\mod\_vti_cnf
```

```
[.]                           [..]                          core.html
directive-dict.html           directives.html               footer.html
header.html                   index.html                    mod_access.html
mod_actions.html              mod_alias.html                mod_asis.html
mod_auth.html                 mod_auth_anon.html            mod_auth_db.html
mod_auth_dbm.html             mod_autoindex.html            mod_browser.html
mod_cern_meta.html            mod_cgi.html                  mod_cookies.html
mod_digest.html               mod_dir.html                  mod_dld.html
mod_dll.html                  mod_env.html                  mod_example.html
mod_expires.html              mod_headers.html              mod_imap.html
mod_include.html              mod_info.html                 mod_isapi.html
mod_log_agent.html            mod_log_common.html           mod_log_config.html
mod_log_referer.html          mod_mime.html                 mod_mime_magic.html
mod_mmap_static.html          mod_negotiation.html          mod_proxy.html
mod_rewrite.html              mod_setenvif.html             mod_so.html
mod_speling.html              mod_status.html               mod_unique_id.html
mod_userdir.html              mod_usertrack.html
              50 File(s)             71,876 bytes


 Directory of D:\Inetpub\wwwroot\manual\vhosts


[.]                           [..]                          details.html
details_1_2.html              examples.html                 fd-limits.html
footer.html                   header.html                   host.html
index.html                    ip-based.html                 name-based.html
vhosts-in-depth.html          virtual-host.html             [_vti_cnf]
              15 File(s)            100,154 bytes


 Directory of D:\Inetpub\wwwroot\manual\vhosts\_vti_cnf


[.]                           [..]                          details.html
details_1_2.html              examples.html                 fd-limits.html
footer.html                   header.html                   host.html
index.html                    ip-based.html                 name-based.html
vhosts-in-depth.html          virtual-host.html
              14 File(s)             10,005 bytes


 Directory of D:\Inetpub\wwwroot\manual\_vti_cnf


[.]                           [..]
bind.html                     cgi_path.html
content-negotiation.html      custom-error.html
dns-caveats.html              dso.html
ebcdic.html                   env.html
footer.html                   handler.html
header.html                   index.html
install.html                  invoking.html
keepalive.html                LICENSE
location.html                 man-template.html
multilogs.html                new_features_1_0.html
new_features_1_1.html         new_features_1_2.html
new_features_1_3.html         process-model.html
sections.html                 sourcereorg.html
stopping.html                 suexec.html
unixware.html                 upgrading_to_1_3.html
windows.html
```

```
           33 File(s)          25,322 bytes

   Directory of D:\Inetpub\wwwroot\Results

[.]              [..]             [classes]        [images]         queen.mid
Results.bak      Results.html     [_vti_cnf]
           8 File(s)          26,195 bytes

   Directory of D:\Inetpub\wwwroot\Results\classes

[.]                       [..]                     [connect]
Results$SymItem.class     Results$SymMouse.class   Results.class
Results.java              [symantec]               [weblogic]
           9 File(s)          89,327 bytes

   Directory of D:\Inetpub\wwwroot\Results\classes\connect

[.]          [..]          [license]    [microsoft] [sybase]
           5 File(s)               0 bytes

   Directory of D:\Inetpub\wwwroot\Results\classes\connect\license

[.]                       [..]                     License.class
LicenseException.class    LicenseManager.class     LicenseRecord.class
           6 File(s)           5,890 bytes

   Directory of D:\Inetpub\wwwroot\Results\classes\connect\microsoft

[.]                       [..]                     MicrosoftDriver.class
           3 File(s)             175 bytes

   Directory of D:\Inetpub\wwwroot\Results\classes\connect\sybase

[.]                       [..]                     SybaseDriver.class
           3 File(s)             166 bytes

   Directory of D:\Inetpub\wwwroot\Results\classes\symantec

[.]          [..]          [beans]    [dti]    [itools] [tools]
           6 File(s)               0 bytes

   Directory of D:\Inetpub\wwwroot\Results\classes\symantec\beans

[.]                       [..]          Beans.class
           3 File(s)             734 bytes

   Directory of D:\Inetpub\wwwroot\Results\classes\symantec\dti

[.]
[..]
BogusComponent.class
ComboBoxDesignTimeInterface.class
ControlDesignTimeInterface.class
CustomPropEditFrame.class
DefaultAwtComponentDesignTimeInterface.class
DefaultAwtPanelDesignTimeInterface.class
EventInfo.class
```

```
EventParamInfo.class
JBlenderFrame.class
JBlenderFrameCreateParams.class
JBlenderMain.class
JBlenderPanel.class
JBlenderPanelGridInfo.class
JBlenderReqQueue.class
JBlenderRequest.class
JClassLoader.class
JCustomPropFrame.class
JDebugStr.class
JInvokeThread.class
NewFont.class
PropInfo.class
PropInfoBoolean.class
PropInfoCharacter.class
PropInfoColor.class
PropInfoCustom.class
PropInfoDouble.class
PropInfoDynamic.class
PropInfoEnum.class
PropInfoFont.class
PropInfoInteger.class
PropInfoInterface.class
PropInfoString.class
PropInfoStringArray.class
PropInfoURL.class
PropMethod.class
Queue.class
ScrollingNames$JPActionHandler.class
ScrollingNames$SymComponent.class
ScrollingNames.class
ScrollingPanelDesignTimeInterface.class
SlideShowDesignTimeInterface.class
StarMain$SymWindow.class
StarMain.class
stars.gif
StdAwtButtonDesignTimeInterface.class
StdAwtCheckboxDesignTimeInterface.class
StdAwtChoiceDesignTimeInterface.class
StdAwtLabelDesignTimeInterface.class
StdAwtScrollbarDesignTimeInterface.class
StdAwtTextAreaDesignTimeInterface.class
StdAwtTextComponentDesignTimeInterface.class
StdAwtTextFieldDesignTimeInterface.class
symName.class
TabPanelDesignTimeInterface.class
TestFrame.class
TreeViewDesignTimeInterface.class
VCFormCompListener$AddRemoveListener.class
VCFormCompListener$Item.class
VCFormCompListener.class
WrappingLabelDesignTimeInterface.class
[_vti_cnf]
             63 File(s)        111,563 bytes

Directory of D:\Inetpub\wwwroot\Results\classes\symantec\dti\_vti_cnf
```

```
[.]    '        [..]           stars.gif
               3 File(s)           265 bytes


Directory of D:\Inetpub\wwwroot\Results\classes\symantec\itools

[.]          [..]         [awt]       [beans]      [db]        [demo]
[lang]       [multimedia] [net]       [util]       [vcafe]
               11 File(s)            0 bytes


Directory of D:\Inetpub\wwwroot\Results\classes\symantec\itools\awt

[.]                                  [..]
AlignStyle.class                     BaseTabbedPanel$Action.class
BaseTabbedPanel$Mouse.class          BaseTabbedPanel.class
BaseTabbedPanelBeanInfo.class        BevelStyle.class
BorderPanel$Veto.class               BorderPanel.class
BorderPanelBeanInfo.class            BorderPanelC16.gif
BorderPanelC32.gif                   ButtonBase$Action.class
ButtonBase$BevelVeto.class           ButtonBase$Focus.class
ButtonBase$FrameVeto.class           ButtonBase$Key.class
ButtonBase$Mouse.class               ButtonBase.class
ButtonBaseBeanInfo.class             ComboBox$Action.class
ComboBox$Focus.class                 ComboBox$Item.class
ComboBox$KeyBox.class                ComboBox$KeyList.class
ComboBox$Mouse.class                 ComboBox.class
ComboBoxBeanInfo.class               ComboBoxC16.gif
ComboBoxC32.gif                      CompareCells.class
CompareFunc.class                    CompareFuncCB.class
DirectionButton$IndntVeto.class      DirectionButton$SizeVeto.class
DirectionButton.class                DirectionButtonBeanInfo.class
DirectionButtonC16.gif               DirectionButtonC32.gif
FormattedTextField$Key.class         FormattedTextField.class
FormattedTextFieldBeanInfo.class     FormattedTextFieldC16.gif
FormattedTextFieldC32.gif            HorizontalSlider$Mouse.class
HorizontalSlider$MouseMtn.class      HorizontalSlider.class
HorizontalSliderBeanInfo.class       HorizontalSliderC16.gif
HorizontalSliderC32.gif              HorizontalSliderThumb.class
HorizontalSliderThumbBot.class       HorizontalSliderThumbBoth.class
HorizontalSliderThumbTop.class       [image]
ImageButton$StyleVeto.class          ImageButton.class
ImageButtonBeanInfo.class            ImageButtonC16.gif
ImageButtonC32.gif                   ImageHTMLLink.class
ImageHTMLLinkBeanInfo.class          ImageHTMLLinkC16.gif
ImageHTMLLinkC32.gif                 ImageListBox$Adjustment.class
ImageListBox$Key.class               ImageListBox$Mouse.class
ImageListBox$MouseMotion.class       ImageListBox.class
ImageListBoxBeanInfo.class           ImageListBoxC16.gif
ImageListBoxC32.gif                  ImagePanel.class
ImagePanelBeanInfo.class             ImagePanelC16.gif
ImagePanelC32.gif                    InvalidTreeNodeException.class
InvisibleButton$Mouse.class          InvisibleButton.class
InvisibleButtonBeanInfo.class        InvisibleButtonC16.gif
InvisibleButtonC32.gif               InvisibleHTMLLink$Action.class
InvisibleHTMLLink$Mouse.class        InvisibleHTMLLink.class
InvisibleHTMLLinkBeanInfo.class      InvisibleHTMLLinkC16.gif
InvisibleHTMLLinkC32.gif             KeyPressManagerPanel$A.class
```

```
KeyPressManagerPanel$K.class              KeyPressManagerPanel.class
KeyPressManagerPanelBeanInfo.class        KeyPressManagerPanelC16.gif
KeyPressManagerPanelC32.gif               Label3D.class
Label3DBeanInfo.class                     Label3DC16.gif
Label3DC32.gif                            LabelButton$HAVeto.class
LabelButton$VAVeto.class                  LabelButton.class
LabelButtonBeanInfo.class                 LabelButtonC16.gif
LabelButtonC32.gif                        LabelHTMLLink.class
LabelHTMLLinkBeanInfo.class               LabelHTMLLinkC16.gif
LabelHTMLLinkC32.gif                       ListItem.class
Matrix.class                              MatrixEnumeration.class
[multiList]                               MultiList$Adjustment.class
MultiList$Focus.class                     MultiList$Key.class
MultiList$Mouse.class                     MultiList$MouseMotion.class
MultiList.class                           MultiListBeanInfo.class
MultiListC16.gif                          MultiListC32.gif
Orientation.class                         RadioButtonGroupPanel.class
RadioButtonGroupPanelBeanInfo.class       RadioButtonGroupPanelC16.gif
RadioButtonGroupPanelC32.gif              RollOverButton$Mouse.class
RollOverButton$MouseMtn.class             RollOverButton.class
RollOverButtonBeanInfo.class              RollOverButtonC16.gif
RollOverButtonC32.gif                     ScrollingPanel.class
ScrollingPanelBeanInfo.class              ScrollingPanelC16.gif
ScrollingPanelC32.gif                     [shape]
Slider.class                              SliderBeanInfo.class
SliderTick.class                          SplitterPanel$Mouse.class
SplitterPanel$MouseMtn.class              SplitterPanel$STVeto.class
SplitterPanel.class                       SplitterPanelBeanInfo.class
SplitterPanelC16.gif                      SplitterPanelC32.gif
StateCheckBox$Mouse.class                 StateCheckBox.class
StateCheckBoxBeanInfo.class               StateCheckBoxC16.gif
StateCheckBoxC32.gif                      StatusBar.class
StatusBarBeanInfo.class                   StatusBarC16.gif
StatusBarC32.gif                          TabPanel$PropertyChange.class
TabPanel.class                            TabPanelBeanInfo.class
TabPanelC16.gif                           TabPanelC32.gif
TransparencyTrick.class                   TransparencyTrickUtils.class
TreeNode.class                            TreeView$Adjustment.class
TreeView$Focus.class                      TreeView$Key.class
TreeView$Mouse.class                      TreeView.class
TreeViewBeanInfo.class                    TreeViewC16.gif
TreeViewC32.gif                           [util]
VerticalSlider$Mouse.class                VerticalSlider$MouseMtn.class
VerticalSlider.class                      VerticalSliderBeanInfo.class
VerticalSliderC16.gif                     VerticalSliderC32.gif
VerticalSliderThumb.class                 VerticalSliderThumbBoth.class
VerticalSliderThumbLeft.class             VerticalSliderThumbRight.class
WrappingLabel.class                       WrappingLabelBeanInfo.class
WrappingLabelC16.gif                      WrappingLabelC32.gif
[_vti_cnf]
             189 File(s)        652,892 bytes


Directory of D:\Inetpub\wwwroot\Results\classes\symantec\itools\awt\image

[.]                    [..]                     DarkenFilter.class
FadeFilter.class       LightenFilter.class
               5 File(s)        7,631 bytes
```

Directory of D:\Inetpub\wwwroot\Results\classes\symantec\itools\awt\multiList

[.]   [..]
Cell.class   CompareTextAndImageCells.class
TextAndImageCell.class
                  5 File(s)   4,490 bytes

Directory of D:\Inetpub\wwwroot\Results\classes\symantec\itools\awt\shape

[.]   [..]
Circle.class   CircleBeanInfo.class
CircleC16.gif   CircleC32.gif
Ellipse.class   EllipseBeanInfo.class
EllipseC16.gif   EllipseC32.gif
HorizontalLine.class   HorizontalLineBeanInfo.class
HorizontalLineC16.gif   HorizontalLineC32.gif
Line.class   LineBeanInfo.class
LineC16.gif   LineC32.gif
Rect.class   RectBeanInfo.class
RectC16.gif   RectC32.gif
Shape.class   ShapeBeanInfo.class
Square.class   SquareBeanInfo.class
SquareC16.gif   SquareC32.gif
VerticalLine.class   VerticalLineBeanInfo.class
VerticalLineC16.gif   VerticalLineC32.gif
[_vti_cnf]
                 33 File(s)   47,004 bytes

Directory of
D:\Inetpub\wwwroot\Results\classes\symantec\itools\awt\shape_vti_cnf

[.]   [..]   CircleC16.gif
CircleC32.gif   EllipseC16.gif   EllipseC32.gif
HorizontalLineC16.gif HorizontalLineC32.gif   LineC16.gif
LineC32.gif   RectC16.gif   RectC32.gif
SquareC16.gif   SquareC32.gif   VerticalLineC16.gif
VerticalLineC32.gif
                 16 File(s)   3,710 bytes

Directory of D:\Inetpub\wwwroot\Results\classes\symantec\itools\awt\util

[.]   [..]
Calendar$Action.class   Calendar$Mouse.class
Calendar.class   CalendarBeanInfo.class
CalendarC16.gif   CalendarC32.gif
ColorUtils.class   [dialog]
[edit]   ProgressBar.class
ProgressBarBeanInfo.class   ProgressBarC16.gif
ProgressBarC32.gif   [spinner]
StatusScroller.class   StatusScrollerBeanInfo.class
StatusScrollerC16.gif   StatusScrollerC32.gif
ToolBarPanel$Veto.class   ToolBarPanel.class
ToolBarPanelBeanInfo.class   ToolBarPanelC16.gif
ToolBarPanelC32.gif   ToolBarSpacer$Veto.class
ToolBarSpacer.class   ToolBarSpacerBeanInfo.class
ToolBarSpacerC16.gif   ToolBarSpacerC32.gif

269

```
Util.class                         [_vti_cnf]
                    32 File(s)      93,195 bytes

   Directory of D:\Inetpub\wwwroot\Results\classes\symantec\itools\awt\util\dialog

[.]                    [..]                     DialogBox.class      ModalDialog.class
                 4 File(s)          2,449 bytes

   Directory of D:\Inetpub\wwwroot\Results\classes\symantec\itools\awt\util\edit

[.]                                          [..]
AddressCode.class                            AddressCodeBeanInfo.class
IntlLongDistPhoneNumber.class                IntlLongDistPhoneNumberBeanInfo.class
IntlLongDistPhoneNumberC16.gif               IntlLongDistPhoneNumberC32.gif
LocalPhoneNumber.class                       LocalPhoneNumberBeanInfo.class
LocalPhoneNumberC16.gif                      LocalPhoneNumberC32.gif
LongDistPhoneNumber.class                    LongDistPhoneNumberBeanInfo.class
LongZipCode.class                            LongZipCodeBeanInfo.class
LongZipCodeC16.gif                           LongZipCodeC32.gif
PhoneNumber.class                            PhoneNumberBeanInfo.class
PostalCode.class                             PostalCodeBeanInfo.class
PostalCodeC16.gif                            PostalCodeC32.gif
SocialIDNumber.class                         SocialIDNumberBeanInfo.class
SocialInsuranceNumber.class                  SocialInsuranceNumberBeanInfo.class
SocialInsuranceNumberC16.gif                 SocialInsuranceNumberC32.gif
SocialSecurityNumber.class                   SocialSecurityNumberBeanInfo.class
SocialSecurityNumberC16.gif                  SocialSecurityNumberC32.gif
USLongDistPhoneNumber.class                  USLongDistPhoneNumberBeanInfo.class
USLongDistPhoneNumberC16.gif                 USLongDistPhoneNumberC32.gif
ZipCode.class                                ZipCodeBeanInfo.class
ZipCodeC16.gif                               ZipCodeC32.gif
[_vti_cnf]
                 43 File(s)         44,980 bytes

   Directory of
D:\Inetpub\wwwroot\Results\classes\symantec\itools\awt\util\edit\_vti_cnf

[.]                                          [..]
IntlLongDistPhoneNumberC16.gif               IntlLongDistPhoneNumberC32.gif
LocalPhoneNumberC16.gif                      LocalPhoneNumberC32.gif
LongZipCodeC16.gif                           LongZipCodeC32.gif
PostalCodeC16.gif                            PostalCodeC32.gif
SocialInsuranceNumberC16.gif                 SocialInsuranceNumberC32.gif
SocialSecurityNumberC16.gif                  SocialSecurityNumberC32.gif
USLongDistPhoneNumberC16.gif                 USLongDistPhoneNumberC32.gif
ZipCodeC16.gif                               ZipCodeC32.gif
                 18 File(s)          4,240 bytes

   Directory of
D:\Inetpub\wwwroot\Results\classes\symantec\itools\awt\util\spinner

[.]                                          [..]
DaySpinner.class                             DaySpinnerBeanInfo.class
DaySpinnerC16.gif                            DaySpinnerC32.gif
HorizontalSpinButtonPanel.class              ListSpinner.class
ListSpinnerBeanInfo.class                    ListSpinnerC16.gif
ListSpinnerC32.gif                           MonthSpinner.class
```

```
MonthSpinnerBeanInfo.class          MonthSpinnerC16.gif
MonthSpinnerC32.gif                 NumericSpinner.class
NumericSpinnerBeanInfo.class        NumericSpinnerC16.gif
NumericSpinnerC32.gif               SpinButtonPanel$Action.class
SpinButtonPanel.class               Spinner$Action.class
Spinner$CurrentVeto.class           Spinner$MaxVeto.class
Spinner$MinVeto.class               Spinner.class
SpinnerBeanInfo.class               VerticalSpinButtonPanel.class
[_vti_cnf]
              29 File(s)          74,491 bytes

 Directory of
D:\Inetpub\wwwroot\Results\classes\symantec\itools\awt\util\spinner\_vti_cnf

[.]                     [..]                    DaySpinnerC16.gif
DaySpinnerC32.gif       ListSpinnerC16.gif      ListSpinnerC32.gif
MonthSpinnerC16.gif     MonthSpinnerC32.gif     NumericSpinnerC16.gif
NumericSpinnerC32.gif
              10 File(s)          2,120 bytes

 Directory of
D:\Inetpub\wwwroot\Results\classes\symantec\itools\awt\util\_vti_cnf

[.]                     [..]                    CalendarC16.gif
CalendarC32.gif         ProgressBarC16.gif      ProgressBarC32.gif
StatusScrollerC16.gif   StatusScrollerC32.gif   ToolBarPanelC16.gif
ToolBarPanelC32.gif     ToolBarSpacerC16.gif    ToolBarSpacerC32.gif
              12 File(s)          2,650 bytes

 Directory of D:\Inetpub\wwwroot\Results\classes\symantec\itools\awt\_vti_cnf

[.]                             [..]
BorderPanelC16.gif              BorderPanelC32.gif
ComboBoxC16.gif                 ComboBoxC32.gif
DirectionButtonC16.gif          DirectionButtonC32.gif
FormattedTextFieldC16.gif       FormattedTextFieldC32.gif
HorizontalSliderC16.gif         HorizontalSliderC32.gif
ImageButtonC16.gif              ImageButtonC32.gif
ImageHTMLLinkC16.gif            ImageHTMLLinkC32.gif
ImageListBoxC16.gif             ImageListBoxC32.gif
ImagePanelC16.gif               ImagePanelC32.gif
InvisibleButtonC16.gif          InvisibleButtonC32.gif
InvisibleHTMLLinkC16.gif        InvisibleHTMLLinkC32.gif
KeyPressManagerPanelC16.gif     KeyPressManagerPanelC32.gif
Label3DC16.gif                  Label3DC32.gif
LabelButtonC16.gif              LabelButtonC32.gif
LabelHTMLLinkC16.gif            LabelHTMLLinkC32.gif
MultiListC16.gif                MultiListC32.gif
RadioButtonGroupPanelC16.gif    RadioButtonGroupPanelC32.gif
RollOverButtonC16.gif           RollOverButtonC32.gif
ScrollingPanelC16.gif           ScrollingPanelC32.gif
SplitterPanelC16.gif            SplitterPanelC32.gif
StateCheckBoxC16.gif            StateCheckBoxC32.gif
StatusBarC16.gif                StatusBarC32.gif
TabPanelC16.gif                 TabPanelC32.gif
TreeViewC16.gif                 TreeViewC32.gif
VerticalSliderC16.gif           VerticalSliderC32.gif
```

```
WrappingLabelC16.gif          WrappingLabelC32.gif
          54 File(s)           13,782 bytes


  Directory of D:\Inetpub\wwwroot\Results\classes\symantec\itools\beans

[.]                           [..]
ConnectionDescriptor.class    PropertyChangeSupport.class
SymantecBeanDescriptor.class  VetoableChangeSupport.class
           6 File(s)            9,948 bytes


  Directory of D:\Inetpub\wwwroot\Results\classes\symantec\itools\db

[.]                           [..]
[awt]                          ConnectionBeanIconColor16.gif
ConnectionBeanIconColor32.gif  DataSourceIconColor16.gif
DataSourceIconColor32.gif      MediatorDSIconColor16.gif
MediatorDSIconColor32.gif      MediatorIconColor16.gif
MediatorIconColor32.gif        [_vti_cnf]
          12 File(s)            4,624 bytes


  Directory of D:\Inetpub\wwwroot\Results\classes\symantec\itools\db\awt

[.]                           [..]
ComboBoxC16.gif                ComboBoxC32.gif
DBTstampC16.gif                DBTstampC32.gif
FormattedTextFieldC16.gif      FormattedTextFieldC32.gif
NervousTextC16.gif             NervousTextC32.gif
StateCheckBoxC16.gif           StateCheckBoxC32.gif
[_vti_cnf]
          13 File(s)      .     2,185 bytes


  Directory of D:\Inetpub\wwwroot\Results\classes\symantec\itools\db\awt\_vti_cnf

[.]            .              [..]
ComboBoxC16.gif                ComboBoxC32.gif
DBTstampC16.gif                DBTstampC32.gif
FormattedTextFieldC16.gif      FormattedTextFieldC32.gif
NervousTextC16.gif             NervousTextC32.gif
StateCheckBoxC16.gif           StateCheckBoxC32.gif
          12 File(s)            2,650 bytes


  Directory of D:\Inetpub\wwwroot\Results\classes\symantec\itools\db\_vti_cnf

[.]                           [..]
ConnectionBeanIconColor16.gif  ConnectionBeanIconColor32.gif
DataSourceIconColor16.gif      DataSourceIconColor32.gif
MediatorDSIconColor16.gif      MediatorDSIconColor32.gif
MediatorIconColor16.gif        MediatorIconColor32.gif
          10 File(s)            2,120 bytes


  Directory of D:\Inetpub\wwwroot\Results\classes\symantec\itools\demo

[.]            [..]            Demo.class       DemoFrame.class
          4 File(s)           5,268 bytes


  Directory of D:\Inetpub\wwwroot\Results\classes\symantec\itools\lang
```

```
[.]                [..]              Context.class   OS.class
                4 File(s)            3,962 bytes


   Directory of D:\Inetpub\wwwroot\Results\classes\symantec\itools\multimedia

[.]                               [..]
Animator.class                    AnimatorBeanInfo.class
AnimatorC16.gif                   AnimatorC32.gif
AnimatorImage.class               Firework.class
FireworkBeanInfo.class            FireworkC16.gif
FireworkC32.gif                   ImageViewer.class
ImageViewerBeanInfo.class         ImageViewerC16.gif
ImageViewerC32.gif                MovingAnimation.class
MovingAnimationBeanInfo.class     MovingAnimationC16.gif
MovingAnimationC32.gif            NervousText.class
NervousTextBeanInfo.class         NervousTextC16.gif
NervousTextC32.gif                Plasma.class
PlasmaBeanInfo.class              PlasmaC16.gif
PlasmaC32.gif                     Rocket.class
ScrollingText$Mouse.class         ScrollingText$MouseMotion.class
ScrollingText.class               ScrollingTextBeanInfo.class
ScrollingTextC16.gif              ScrollingTextC32.gif
SlideShow.class                   SoundPlayer.class
SoundPlayerBeanInfo.class         SoundPlayerC16.gif
SoundPlayerC32.gif                SoundViewerItem.class
SoundViewerThread.class           [_vti_cnf]
                42 File(s)        124,452 bytes


   Directory of
D:\Inetpub\wwwroot\Results\classes\symantec\itools\multimedia\_vti_cnf

[.]                     [..]                    AnimatorC16.gif
AnimatorC32.gif         FireworkC16.gif         FireworkC32.gif
ImageViewerC16.gif      ImageViewerC32.gif      MovingAnimationC16.gif
MovingAnimationC32.gif  NervousTextC16.gif      NervousTextC32.gif
PlasmaC16.gif           PlasmaC32.gif           ScrollingTextC16.gif
ScrollingTextC32.gif    SoundPlayerC16.gif      SoundPlayerC32.gif
                18 File(s)              4,240 bytes


   Directory of D:\Inetpub\wwwroot\Results\classes\symantec\itools\net

[.]                [..]                    RelativeURL.class
                3 File(s)            1,619 bytes


   Directory of D:\Inetpub\wwwroot\Results\classes\symantec\itools\util

[.]                [..]                    GeneralUtils.class
Timer.class        ZOrderUtils.class
                5 File(s)            9,479 bytes


   Directory of D:\Inetpub\wwwroot\Results\classes\symantec\itools\vcafe

[.]                               [..]
[beanhelp]                        Bundler$Action.class
Bundler$Item.class                Bundler$Key.class
Bundler$SymText.class             Bundler$Window.class
Bundler.class                     BundlerNode.class
```

```
[commtool]                            DependentDialog$Action.class
DependentDialog$Key.class             DependentDialog$Window.class
DependentDialog.class                 DependentItem.class
ErsatzFile.class                      InternalMapper.class
              18 File(s)                37,755 bytes

     Directory of D:\Inetpub\wwwroot\Results\classes\symantec\itools\vcafe\beanhelp

[.]                                   [..]
BackRunContainer.class                BackRunMemberInfo.class
BackRunMenu.class                     BackRunMenuBar.class
BackrunObjectCreator.class            BackRunParent.class
[beaninfo]                            ClassInfo.class
ConvertToVPO$Helper.class             ConvertToVPO.class
DoOnBean.class                        [editors]
ImageData.class                       InputStreamProducer.class
JarClassInfo.class                    JarInfo.class
JarManager.class                      LoadedJar.class
Manifest.class                        MessageHeader.class
ObjectInfo.class                      ObjInputStreamWithLoader.class
SimpleClassLoader.class               [simpleresource]
              26 File(s)                77,551 bytes

     Directory of
D:\Inetpub\wwwroot\Results\classes\symantec\itools\vcafe\beanhelp\beaninfo

[.]                                   [..]
CanvasBeanInfo.class                  ComponentBeanInfo.class
ContainerBeanInfo.class               PanelBeanInfo.class
TextComponentBeanInfo.class           TextFieldBeanInfo.class
               8 File(s)                24,595 bytes

     Directory of
D:\Inetpub\wwwroot\Results\classes\symantec\itools\vcafe\beanhelp\editors

[.]                                   [..]
BooleanEditor.class                   CharacterEditor.class
CheckboxGroupEditor.class             ColorEditor.class
CursorEditor.class                    FontEditor.class
IntegerEditor.class                   LayoutManagerEditor.class
RectangleEditor.class                 StringArrayEditor.class
StringEditor.class                    SymPropertyEditorSupport.class
URLArrayEditor.class                  URLEditor.class
UseParentsBackgroundEditor.class      UseParentsFontEditor.class
UseParentsForegroundEditor.class
              19 File(s)               18,547 bytes

     Directory of
D:\Inetpub\wwwroot\Results\classes\symantec\itools\vcafe\beanhelp\simpleresource

[.]                                   [..]
Handler.class                         SimpleResourceConnection.class
               4 File(s)                3,042 bytes

     Directory of D:\Inetpub\wwwroot\Results\classes\symantec\itools\vcafe\commtool

[.]                                   [..]
```

```
Memory.class                MeterTool$1.class
MeterTool$SymItem.class      MeterTool$SymWindow.class
MeterTool.class
              7 File(s)           7,771 bytes

 Directory of D:\Inetpub\wwwroot\Results\classes\symantec\tools

[.]         [..]        [debug]    [java]    [javac]    [ttydebug]
              6 File(s)            0 bytes

 Directory of D:\Inetpub\wwwroot\Results\classes\symantec\tools\debug

[.]
[..]
Agent.class
AgentConstants.class
AgentIn.class
AgentOutputStream.class
BreakpointHandler.class
BreakpointQueue.class
BreakpointSet.class
DebuggerCallback.class
EmptyApp.class
ExecReRouter.class
ExpressionOutOfContextException.class
ExpressionWrongNumberLocalVarsException.class
Field.class
InvalidPCException.class
LineNumber.class
LocalCommInputStream.class
LocalCommOutputStream.class
LocalCommStream.class
LocalVariable.class
MainThread.class
NoSessionException.class
NoSuchFieldException.class
NoSuchLineNumberException.class
RemoteAgent.class
RemoteArray.class
RemoteBoolean.class
RemoteByte.class
RemoteChar.class
RemoteClass.class
RemoteDebugger.class
RemoteDouble.class
RemoteField.class
RemoteFloat.class
RemoteInt.class
RemoteLong.class
RemoteObject.class
RemoteShort.class
RemoteStackFrame.class
RemoteStackVariable.class
RemoteString.class
RemoteThread.class
RemoteThreadGroup.class
RemoteValue.class
```

```
ResponseStream.class
StackFrame.class
SymDbgExec.class
SymDbgExprFindContext.class
SymDbgExprHandler.class
SymDbgExprResult.class
SymDbgLoadExpr.class
SymDbgNotify.class
SymDbgProcess.class
ThreadList.class
               55 File(s)          136,888 bytes

  Directory of D:\Inetpub\wwwroot\Results\classes\symantec\tools\java

[.]                              [..]
VEDbgJavaDataRec.class           VEDbgJavaDebug.class
VEDbgJavaIntf.class              VEDbgJavaKernel.class
VEDbgJavaMsgRouter.class         VEDbgJavaRemoteApp.class
VEDbgJavaResource.class          VEDbgJavaResource_ja.class
VEDbgJavaRuntimeCmd.class
               11 File(s)          39,759 bytes

  Directory of D:\Inetpub\wwwroot\Results\classes\symantec\tools\javac

[.]                       [..]                    BatchEnvironment.class
BatchParser.class         ErrorConsumer.class     ErrorMessage.class
JavaIDDE.class            Main.class
               8 File(s)          33,103 bytes

  Directory of D:\Inetpub\wwwroot\Results\classes\symantec\tools\ttydebug

[.]            [..]          TTY.class
               3 File(s)          28,429 bytes

  Directory of D:\Inetpub\wwwroot\Results\classes\weblogic

[.]   [..]    [jdbc]
               3 File(s)                  0 bytes

  Directory of D:\Inetpub\wwwroot\Results\classes\weblogic\jdbc

[.]             [..]              [common4]    [microsoft4] [sybase4]
               5 File(s)                  0 bytes

  Directory of D:\Inetpub\wwwroot\Results\classes\weblogic\jdbc\common4

[.]             [..]              [compression]  [cryptography] [io]
[net]           [sql]             [util]
               8 File(s)                  0 bytes

  Directory of
D:\Inetpub\wwwroot\Results\classes\weblogic\jdbc\common4\compression

[.]                       [..]                    Compressor.class
LzssCompressor.class      LzssNode.class          TdsCompressor.class
               6 File(s)          3,880 bytes
```

```
Directory of
D:\Inetpub\wwwroot\Results\classes\weblogic\jdbc\common4\cryptography

[.]                             [..]
BlowfishCryptographer.class     Cryptographer.class
DESCryptographer.class          XorCryptographer.class
            6 File(s)           38,039 bytes


Directory of D:\Inetpub\wwwroot\Results\classes\weblogic\jdbc\common4\io

[.]                                 [..]
AsciiInputStream.class              BufferedFilterInputStream.class
BufferedFilterOutputStream.class    CompressedInputStream.class
CompressedOutputStream.class        EncryptedInputStream.class
EncryptedOutputStream.class         HexInputStream.class
UnicodeInputStream.class
            11 File(s)              10,326 bytes


Directory of D:\Inetpub\wwwroot\Results\classes\weblogic\jdbc\common4\net

[.]                     [..]                    SocketExtras.class
SocketTimeout.class
            4 File(s)           4,352 bytes


Directory of D:\Inetpub\wwwroot\Results\classes\weblogic\jdbc\common4\sql

[.]                                 [..]
AsyncWriter.class                   BaseConnection.class
BaseHandler.class                   BaseString.class
ConnectDriver.class                 ConnectionInfo.class
JdbcExpandedSqlParser.class         MicrosoftConnection.class
MicrosoftDatabaseMetaData.class     MicrosoftLoginStatement.class
TdsCallableStatement.class          TdsColumnMetaData.class
TdsConnection.class                 TdsDatabaseMetaData.class
TdsEntry.class                      TdsException.class
TdsInputStream.class                TdsLoginStatement.class
TdsOutputStream.class               TdsParameter.class
TdsPreparedInputStream.class        TdsPreparedStatement.class
TdsResultSet.class                  TdsResultSetMetaData.class
TdsStatement.class
            27 File(s)          142,174 bytes


Directory of D:\Inetpub\wwwroot\Results\classes\weblogic\jdbc\common4\util

[.]                 [..]                BitsArray.class    Day.class
            4 File(s)           2,822 bytes


Directory of D:\Inetpub\wwwroot\Results\classes\weblogic\jdbc\microsoft4

[.]             [..]            Driver.class
            3 File(s)           859 bytes


Directory of D:\Inetpub\wwwroot\Results\classes\weblogic\jdbc\sybase4

[.]             [..]            Driver.class
            3 File(s)           841 bytes
```

```
Directory of D:\Inetpub\wwwroot\Results\images

[.]          [..]           help.gif
             3 File(s)                276 bytes

Directory of D:\Inetpub\wwwroot\Results\_vti_cnf

[.]              [..]            Results.html
                 3 File(s)               533 bytes

Directory of D:\Inetpub\wwwroot\secureSurvey

[.]          [..]         ADOVBS.inc   [images]      index.asp      [javascript]
[sounds]     submit.asp   submit.BAK
             9 File(s)             117,259 bytes

Directory of D:\Inetpub\wwwroot\secureSurvey\images

[.]          [..]          COMPASS.GIF    Connie.jpg     DENIM.JPG
DESERT.GIF   FLY.GIF       help.gif       IND-DECK.GIF   IND-STRN.GIF
RADAR.GIF    [_vti_cnf]
             12 File(s)          240,069 bytes

Directory of D:\Inetpub\wwwroot\secureSurvey\images\_vti_cnf

[.]          [..]          COMPASS.GIF    Connie.jpg     DENIM.JPG
DESERT.GIF   FLY.GIF       IND-DECK.GIF   IND-STRN.GIF   RADAR.GIF
             10 File(s)           2,253 bytes

Directory of D:\Inetpub\wwwroot\secureSurvey\javascript

[.]              [..]          checksurvey.js    clearsurvey.js
[_vti_cnf]
             5 File(s)           9,549 bytes

Directory of D:\Inetpub\wwwroot\secureSurvey\javascript\_vti_cnf

[.]              [..]            checksurvey.js
             3 File(s)               117 bytes

Directory of D:\Inetpub\wwwroot\secureSurvey\sounds

[.]          [..]          bluesbro.mid   GONNAFLY.MID   mission.mid
SAWYER.MID   viewkill.mid
             7 File(s)           196,659 bytes

Directory of D:\Inetpub\wwwroot\security

[.]              [..]            disclaimer.html
             3 File(s)            2,047 bytes

Directory of D:\Inetpub\wwwroot\servletrunner

[.]          [..]          BACKUP        [BIN]          DeIsL1.isu
[DOC]        [EXAMPLES]    fortune.log   [LIB]          README
[SRC]        SURVEY.TXT
             12 File(s)           13,030 bytes
```

Directory of D:\Inetpub\wwwroot\servletrunner\BIN

[.]   [..]   servletrunner.exe
            3 File(s)   41,984 bytes

Directory of D:\Inetpub\wwwroot\servletrunner\DOC

[.]   [..]   [APIDOC]
[IMAGES]   servlet_tutorial.html   [_vti_cnf]
            6 File(s)   39,842 bytes

Directory of D:\Inetpub\wwwroot\servletrunner\DOC\APIDOC

[.]
[..]
AllNames.html
[IMAGES]
javax.servlet.GenericServlet.html
javax.servlet.http.Cookie.html
javax.servlet.http.HttpServlet.html
javax.servlet.http.HttpServletRequest.html
javax.servlet.http.HttpServletResponse.html
javax.servlet.http.HttpSession.html
javax.servlet.http.HttpSessionBindingEvent.html
javax.servlet.http.HttpSessionBindingListener.html
javax.servlet.http.HttpSessionContext.html
javax.servlet.http.HttpUtils.html
javax.servlet.Servlet.html
javax.servlet.ServletConfig.html
javax.servlet.ServletContext.html
javax.servlet.ServletException.html
javax.servlet.ServletInputStream.html
javax.servlet.ServletOutputStream.html
javax.servlet.ServletRequest.html
javax.servlet.ServletResponse.html
javax.servlet.SingleThreadModel.html
javax.servlet.UnavailableException.html
Package-javax.servlet.html
Package-javax.servlet.http.html
packages.html
tree.html
[_vti_cnf]
            29 File(s)   279,098 bytes

Directory of D:\Inetpub\wwwroot\servletrunner\DOC\APIDOC\IMAGES

[.]   [..]   blue-ball-small.gif
blue-ball.gif   class-index.gif   constructor-index.gif
constructors.gif   cyan-ball-small.gif   cyan-ball.gif
error-index.gif   exception-index.gif   green-ball-small.gif
green-ball.gif   interface-index.gif   [JAVA.AWT]
magenta-ball-small.gif   magenta-ball.gif   method-index.gif
METHODS.GIF   OpenBookIcon.gif   package-index.gif
red-ball-small.gif   RED-BALL.GIF   variable-index.gif
variables.gif   yellow-ball-small.gif   yellow-ball.gif
[_vti_cnf]

```
                28 File(s)            25,671 bytes

    Directory of D:\Inetpub\wwwroot\servletrunner\DOC\APIDOC\IMAGES\JAVA.AWT

[.]                  [..]              GridBagEx.gif     [_vti_cnf]
                 4 File(s)            2,453 bytes

    Directory of
D:\Inetpub\wwwroot\servletrunner\DOC\APIDOC\IMAGES\JAVA.AWT\_vti_cnf

[.]                  [..]              GridBagEx.gif
                 3 File(s)              266 bytes

    Directory of D:\Inetpub\wwwroot\servletrunner\DOC\APIDOC\IMAGES\_vti_cnf

[.]                       [..]                    blue-ball-small.gif
blue-ball.gif             class-index.gif         constructor-index.gif
constructors.gif          cyan-ball-small.gif     cyan-ball.gif
error-index.gif           exception-index.gif     green-ball-small.gif
green-ball.gif            interface-index.gif     magenta-ball-small.gif
magenta-ball.gif          method-index.gif        METHODS.GIF
OpenBookIcon.gif          package-index.gif       red-ball-small.gif
RED-BALL.GIF              variable-index.gif       variables.gif
yellow-ball-small.gif     yellow-ball.gif
                26 File(s)            14,099 bytes

    Directory of D:\Inetpub\wwwroot\servletrunner\DOC\APIDOC\_vti_cnf

[.]
[..]
AllNames.html
javax.servlet.GenericServlet.html
javax.servlet.http.Cookie.html
javax.servlet.http.HttpServlet.html
javax.servlet.http.HttpServletRequest.html
javax.servlet.http.HttpServletResponse.html
javax.servlet.http.HttpSession.html
javax.servlet.http.HttpSessionBindingEvent.html
javax.servlet.http.HttpSessionBindingListener.html
javax.servlet.http.HttpSessionContext.html
javax.servlet.http.HttpUtils.html
javax.servlet.Servlet.html
javax.servlet.ServletConfig.html
javax.servlet.ServletContext.html
javax.servlet.ServletException.html
javax.servlet.ServletInputStream.html
javax.servlet.ServletOutputStream.html
javax.servlet.ServletRequest.html
javax.servlet.ServletResponse.html
javax.servlet.SingleThreadModel.html
javax.servlet.UnavailableException.html
Package-javax.servlet.html
Package-javax.servlet.http.html
packages.html
tree.html
                27 File(s)            72,291 bytes
```

```
Directory of D:\Inetpub\wwwroot\servletrunner\DOC\IMAGES

[.]                    [..]                    BANNER.GIF           survey-reply.gif
[_vti_cnf]
                5 File(s)           10,355 bytes

Directory of D:\Inetpub\wwwroot\servletrunner\DOC\IMAGES\_vti_cnf

[.]                    [..]                    BANNER.GIF           survey-reply.gif
                4 File(s)            654 bytes

Directory of D:\Inetpub\wwwroot\servletrunner\DOC\_vti_cnf

[.]                              [..]                        servlet_tutorial.html
                3 File(s)         2,397 bytes

Directory of D:\Inetpub\wwwroot\servletrunner\EXAMPLES

[.]                    [..]                    JdcSurvey.html
SessionServlet.class   SessionServlet.java     SimpleServlet.class
SimpleServlet.java     SnoopServlet.class      SnoopServlet.java
[_vti_cnf]
                10 File(s)          20,066 bytes

Directory of D:\Inetpub\wwwroot\servletrunner\EXAMPLES\_vti_cnf

[.]                    [..]              JdcSurvey.html
                3 File(s)             430 bytes

Directory of D:\Inetpub\wwwroot\servletrunner\LIB

[.]          [..]          JSDK.JAR
                3 File(s)          75,743 bytes

Directory of D:\Inetpub\wwwroot\servletrunner\SRC

[.]       [..]     [JAVAX]
                3 File(s)                  0 bytes

Directory of D:\Inetpub\wwwroot\servletrunner\SRC\JAVAX

[.]          [..]          [SERVLET]
                3 File(s)                  0 bytes

Directory of D:\Inetpub\wwwroot\servletrunner\SRC\JAVAX\SERVLET

[.]                            [..]
GenericServlet.java            [HTTP]
Servlet.java                   ServletConfig.java
ServletContext.java            ServletException.java
ServletInputStream.java        ServletOutputStream.java
ServletRequest.java            ServletResponse.java
SingleThreadModel.java         UnavailableException.java
           14 File(s)          55,979 bytes

Directory of D:\Inetpub\wwwroot\servletrunner\SRC\JAVAX\SERVLET\HTTP
```

```
[.]                                   [..]
Cookie.java                           HttpServlet.java
HttpServletRequest.java               HttpServletResponse.java
HttpSession.java                      HttpSessionBindingEvent.java
HttpSessionBindingListener.java       HttpSessionContext.java
HttpUtils.java
               11 File(s)             85,816 bytes


 Directory of D:\Inetpub\wwwroot\SERVLETS

[.]                                   [..]
AllNames.html                         [CONNECT]
[file_utilities]                      FORTUNE.LOG
FortuneClientServlet.class            FortuneClientServlet.html
FortuneClientServlet.java             fortunes.ldb
fortunes.mdb                          FortuneServer.class
FortuneServer.html                    FortuneServer.java
[images]                              packages.html
QuoteSearch.class                     QuoteSearch.html
QuoteSearch.java                      RequestProcessor.class
RequestProcessor.html                 RequestProcessor.java
servlet.properties                    startserver.bat
SURVEY.TXT                            SurveyServlet.class
SurveyServlet.java                    tree.html
[WEBLOGIC]                            WriteToFile.class
WriteToFile.html                      WriteToFile.java
               32 File(s)             3,042,391 bytes


 Directory of D:\Inetpub\wwwroot\SERVLETS\CONNECT

[.]          [..]          [LICENSE]    [microsoft] [SYBASE]
               5 File(s)                0 bytes


 Directory of D:\Inetpub\wwwroot\SERVLETS\CONNECT\LICENSE

[.]                     [..]                         License.class
LicenseException.class  LicenseManager.class         LicenseRecord.class
               6 File(s)             5,890 bytes


 Directory of D:\Inetpub\wwwroot\SERVLETS\CONNECT\microsoft

[.]                     [..]                     MicrosoftDriver.class
               3 File(s)             175 bytes


 Directory of D:\Inetpub\wwwroot\SERVLETS\CONNECT\SYBASE

[.]                     [..]                     SybaseDriver.class
               3 File(s)             166 bytes


 Directory of D:\Inetpub\wwwroot\SERVLETS\file_utilities

[.]                     [..]              WriteToFile.BAK      WriteToFile.class
WriteToFile.java        WriteToFile.~jav
               6 File(s)             8,151 bytes


 Directory of D:\Inetpub\wwwroot\SERVLETS\images
```

```
[.]                    [..]                   blue-ball-small.gif
blue-ball.gif          constructor-index.gif  constructors.gif
green-ball-small.gif   green-ball.gif         magenta-ball.gif
method-index.gif       methods.gif            red-ball-small.gif
red-ball.gif           speaking.gif           variable-index.gif
variables.gif          yellow-ball-small.gif  yellow-ball.gif
          18 File(s)            14,696 bytes


 Directory of D:\Inetpub\wwwroot\SERVLETS\WEBLOGIC

[.]     [..]    [JDBC]
           3 File(s)                0 bytes


 Directory of D:\Inetpub\wwwroot\SERVLETS\WEBLOGIC\JDBC

[.]             [..]             [COMMON4]     [microsoft4] [SYBASE4]
           5 File(s)                0 bytes


 Directory of D:\Inetpub\wwwroot\SERVLETS\WEBLOGIC\JDBC\COMMON4

[.]             [..]             [compression]  [cryptography] [IO]
[NET]           [SQL]            [UTIL]
           8 File(s)                0 bytes


 Directory of D:\Inetpub\wwwroot\SERVLETS\WEBLOGIC\JDBC\COMMON4\compression

[.]                    [..]                    Compressor.class
LzssCompressor.class   LzssNode.class          TdsCompressor.class
           6 File(s)             3,880 bytes


 Directory of D:\Inetpub\wwwroot\SERVLETS\WEBLOGIC\JDBC\COMMON4\cryptography

[.]                         [..]
BlowfishCryptographer.class Cryptographer.class
DESCryptographer.class      XorCryptographer.class
           6 File(s)            38,039 bytes


 Directory of D:\Inetpub\wwwroot\SERVLETS\WEBLOGIC\JDBC\COMMON4\IO

[.]                           [..]
AsciiInputStream.class        BufferedFilterInputStream.class
BufferedFilterOutputStream.class CompressedInputStream.class
CompressedOutputStream.class  EncryptedInputStream.class
EncryptedOutputStream.class   HexInputStream.class
UnicodeInputStream.class
           11 File(s)            10,326 bytes


 Directory of D:\Inetpub\wwwroot\SERVLETS\WEBLOGIC\JDBC\COMMON4\NET

[.]                      [..]                   SocketExtras.class
SocketTimeout.class
           4 File(s)             4,352 bytes


 Directory of D:\Inetpub\wwwroot\SERVLETS\WEBLOGIC\JDBC\COMMON4\SQL

[.]                      [..]
AsyncWriter.class        BaseConnection.class
```

```
BaseHandler.class                      BaseString.class
ConnectDriver.class                    ConnectionInfo.class
JdbcExpandedSqlParser.class            MicrosoftConnection.class
MicrosoftDatabaseMetaData.class        MicrosoftLoginStatement.class
TdsCallableStatement.class             TdsColumnMetaData.class
TdsConnection.class                    TdsDatabaseMetaData.class
TdsEntry.class                         TdsException.class
TdsInputStream.class                   TdsLoginStatement.class
TdsOutputStream.class                  TdsParameter.class
TdsPreparedInputStream.class           TdsPreparedStatement.class
TdsResultSet.class                     TdsResultSetMetaData.class
TdsStatement.class
              27 File(s)        142,174 bytes

  Directory of D:\Inetpub\wwwroot\SERVLETS\WEBLOGIC\JDBC\COMMON4\UTIL

[.]                   [..]                  BitsArray.class    Day.class
              4 File(s)        2,822 bytes

  Directory of D:\Inetpub\wwwroot\SERVLETS\WEBLOGIC\JDBC\microsoft4

[.]                   [..]                  Driver.class
              3 File(s)          859 bytes

  Directory of D:\Inetpub\wwwroot\SERVLETS\WEBLOGIC\JDBC\SYBASE4

[.]                   [..]                  Driver.class
              3 File(s)          841 bytes

  Directory of D:\Inetpub\wwwroot\sounds

[.]                   [..]                  bluesbro.mid    GONNAFLY.MID    mission.mid
SAWYER.MID            viewkill.mid
              7 File(s)        196,659 bytes

  Directory of D:\Inetpub\wwwroot\Survey

[.]                   [..]                  Applet.html    [classes]        [images]
[javascript]          [sounds]              Survey.html    [_vti_cnf]
              9 File(s)         97,431 bytes

  Directory of D:\Inetpub\wwwroot\Survey\classes

[.]                                    [..]
AboutDialog$SymAction.class            AboutDialog$SymWindow.class
AboutDialog.class                      AboutDialog.java
Applet1$SymMouse.class                 [connect]
Questionnaire$SymAction.class          Questionnaire$SymMouse.class
Questionnaire$SymWindow.class          Questionnaire.class
Questionnaire.java                     QuitDialog$SymAction.class
QuitDialog$SymWindow.class             QuitDialog.class
QuitDialog.java                        Survey$SymMouse.class
Survey.class                           Survey.java
[symantec]                             UserInput$SymMouse.class
UserInput$SymWindow.class              UserInput.class
UserInput.java                         [weblogic]
              26 File(s)       126,073 bytes
```

```
Directory of D:\Inetpub\wwwroot\Survey\classes\connect

[.]           [..]          [license]   [microsoft] [sybase]
          5 File(s)              0 bytes

 Directory of D:\Inetpub\wwwroot\Survey\classes\connect\license

[.]                    [..]                        License.class
LicenseException.class  LicenseManager.class      LicenseRecord.class
          6 File(s)          5,890 bytes

 Directory of D:\Inetpub\wwwroot\Survey\classes\connect\microsoft

[.]                    [..]                        MicrosoftDriver.class
          3 File(s)            175 bytes

 Directory of D:\Inetpub\wwwroot\Survey\classes\connect\sybase

[.]                   [..]                        SybaseDriver.class
          3 File(s)            166 bytes

 Directory of D:\Inetpub\wwwroot\Survey\classes\symantec

[.]        [..]       [beans]   [dti]     [itools] [tools]
          6 File(s)              0 bytes

 Directory of D:\Inetpub\wwwroot\Survey\classes\symantec\beans

[.]           [..]           Beans.class
          3 File(s)            734 bytes

 Directory of D:\Inetpub\wwwroot\Survey\classes\symantec\dti

[.]
[..]
BogusComponent.class
ComboBoxDesignTimeInterface.class
ControlDesignTimeInterface.class
CustomPropEditFrame.class
DefaultAwtComponentDesignTimeInterface.class
DefaultAwtPanelDesignTimeInterface.class
EventInfo.class
EventParamInfo.class
JBlenderFrame.class
JBlenderFrameCreateParams.class
JBlenderMain.class
JBlenderPanel.class
JBlenderPanelGridInfo.class
JBlenderReqQueue.class
JBlenderRequest.class
JClassLoader.class
JCustomPropFrame.class
JDebugStr.class
JInvokeThread.class
NewFont.class
PropInfo.class
```

```
PropInfoBoolean.class
PropInfoCharacter.class
PropInfoColor.class
PropInfoCustom.class
PropInfoDouble.class
PropInfoDynamic.class
PropInfoEnum.class
PropInfoFont.class
PropInfoInteger.class
PropInfoInterface.class
PropInfoString.class
PropInfoStringArray.class
PropInfoURL.class
PropMethod.class
Queue.class
ScrollingNames$JPActionHandler.class
ScrollingNames$SymComponent.class
ScrollingNames.class
ScrollingPanelDesignTimeInterface.class
SlideShowDesignTimeInterface.class
StarMain$SymWindow.class
StarMain.class
stars.gif
StdAwtButtonDesignTimeInterface.class
StdAwtCheckboxDesignTimeInterface.class
StdAwtChoiceDesignTimeInterface.class
StdAwtLabelDesignTimeInterface.class
StdAwtScrollbarDesignTimeInterface.class
StdAwtTextAreaDesignTimeInterface.class
StdAwtTextComponentDesignTimeInterface.class
StdAwtTextFieldDesignTimeInterface.class
symName.class
TabPanelDesignTimeInterface.class
TestFrame.class
TreeViewDesignTimeInterface.class
VCFormCompListener$AddRemoveListener.class
VCFormCompListener$Item.class
VCFormCompListener.class
WrappingLabelDesignTimeInterface.class
[_vti_cnf]
              63 File(s)          111,563 bytes

 Directory of D:\Inetpub\wwwroot\Survey\classes\symantec\dti\_vti_cnf

[.]            [..]           stars.gif
               3 File(s)             265 bytes

 Directory of D:\Inetpub\wwwroot\Survey\classes\symantec\itools

[.]            [..]          [awt]       [beans]     [db]        [demo]
[lang]         [multimedia]  [net]       [util]      [vcafe]
               11 File(s)            0 bytes

 Directory of D:\Inetpub\wwwroot\Survey\classes\symantec\itools\awt

[.]                                      [..]
AlignStyle.class                         BaseTabbedPanel$Action.class
```

```
BaseTabbedPanel$Mouse.class               BaseTabbedPanel.class
BaseTabbedPanelBeanInfo.class             BevelStyle.class
BorderPanel$Veto.class                    BorderPanel.class
BorderPanelBeanInfo.class                 BorderPanelC16.gif
BorderPanelC32.gif                        ButtonBase$Action.class
ButtonBase$BevelVeto.class                ButtonBase$Focus.class
ButtonBase$FrameVeto.class                ButtonBase$Key.class
ButtonBase$Mouse.class                    ButtonBase.class
ButtonBaseBeanInfo.class                  ComboBox$Action.class
ComboBox$Focus.class                      ComboBox$Item.class
ComboBox$KeyBox.class                      ComboBox$KeyList.class
ComboBox$Mouse.class                       ComboBox.class
ComboBoxBeanInfo.class                     ComboBoxC16.gif
ComboBoxC32.gif                            CompareCells.class
CompareFunc.class                          CompareFuncCB.class
DirectionButton$IndntVeto.class            DirectionButton$SizeVeto.class
DirectionButton.class                      DirectionButtonBeanInfo.class
DirectionButtonC16.gif                     DirectionButtonC32.gif
FormattedTextField$Key.class               FormattedTextField.class
FormattedTextFieldBeanInfo.class           FormattedTextFieldC16.gif
FormattedTextFieldC32.gif                  HorizontalSlider$Mouse.class
HorizontalSlider$MouseMtn.class            HorizontalSlider.class
HorizontalSliderBeanInfo.class             HorizontalSliderC16.gif
HorizontalSliderC32.gif                    HorizontalSliderThumb.class
HorizontalSliderThumbBot.class             HorizontalSliderThumbBoth.class
HorizontalSliderThumbTop.class             [image]
ImageButton$StyleVeto.class                ImageButton.class
ImageButtonBeanInfo.class                  ImageButtonC16.gif
ImageButtonC32.gif                         ImageHTMLLink.class
ImageHTMLLinkBeanInfo.class                ImageHTMLLinkC16.gif
ImageHTMLLinkC32.gif                       ImageListBox$Adjustment.class
ImageListBox$Key.class                     ImageListBox$Mouse.class
ImageListBox$MouseMotion.class             ImageListBox.class
ImageListBoxBeanInfo.class                 ImageListBoxC16.gif
ImageListBoxC32.gif                        ImagePanel.class
ImagePanelBeanInfo.class                   ImagePanelC16.gif
ImagePanelC32.gif                          InvalidTreeNodeException.class
InvisibleButton$Mouse.class                InvisibleButton.class
InvisibleButtonBeanInfo.class              InvisibleButtonC16.gif
InvisibleButtonC32.gif                     InvisibleHTMLLink$Action.class
InvisibleHTMLLink$Mouse.class              InvisibleHTMLLink.class
InvisibleHTMLLinkBeanInfo.class            InvisibleHTMLLinkC16.gif
InvisibleHTMLLinkC32.gif                   KeyPressManagerPanel$A.class
KeyPressManagerPanel$K.class               KeyPressManagerPanel.class
KeyPressManagerPanelBeanInfo.class         KeyPressManagerPanelC16.gif
KeyPressManagerPanelC32.gif                Label3D.class
Label3DBeanInfo.class                      Label3DC16.gif
Label3DC32.gif                             LabelButton$HAVeto.class
LabelButton$VAVeto.class                   LabelButton.class
LabelButtonBeanInfo.class                  LabelButtonC16.gif
LabelButtonC32.gif                         LabelHTMLLink.class
LabelHTMLLinkBeanInfo.class                LabelHTMLLinkC16.gif
LabelHTMLLinkC32.gif                       ListItem.class
Matrix.class                               MatrixEnumeration.class
[multiList]                                MultiList$Adjustment.class
MultiList$Focus.class                      MultiList$Key.class
MultiList$Mouse.class                      MultiList$MouseMotion.class
```

```
MultiList.class                        MultiListBeanInfo.class
MultiListC16.gif                       MultiListC32.gif
Orientation.class                      RadioButtonGroupPanel.class
RadioButtonGroupPanelBeanInfo.class    RadioButtonGroupPanelC16.gif
RadioButtonGroupPanelC32.gif           RollOverButton$Mouse.class
RollOverButton$MouseMtn.class          RollOverButton.class
RollOverButtonBeanInfo.class           RollOverButtonC16.gif
RollOverButtonC32.gif                  ScrollingPanel.class
ScrollingPanelBeanInfo.class           ScrollingPanelC16.gif
ScrollingPanelC32.gif                  [shape]
Slider.class                           SliderBeanInfo.class
SliderTick.class                       SplitterPanel$Mouse.class
SplitterPanel$MouseMtn.class           SplitterPanel$STVeto.class
SplitterPanel.class                    SplitterPanelBeanInfo.class
SplitterPanelC16.gif                   SplitterPanelC32.gif
StateCheckBox$Mouse.class              StateCheckBox.class
StateCheckBoxBeanInfo.class            StateCheckBoxC16.gif
StateCheckBoxC32.gif                   StatusBar.class
StatusBarBeanInfo.class                StatusBarC16.gif
StatusBarC32.gif                       TabPanel$PropertyChange.class
TabPanel.class                         TabPanelBeanInfo.class
TabPanelC16.gif                        TabPanelC32.gif
TransparencyTrick.class                TransparencyTrickUtils.class
TreeNode.class                         TreeView$Adjustment.class
TreeView$Focus.class                   TreeView$Key.class
TreeView$Mouse.class                   TreeView.class
TreeViewBeanInfo.class                 TreeViewC16.gif
TreeViewC32.gif                        [util]
VerticalSlider$Mouse.class             VerticalSlider$MouseMtn.class
VerticalSlider.class                   VerticalSliderBeanInfo.class
VerticalSliderC16.gif                  VerticalSliderC32.gif
VerticalSliderThumb.class              VerticalSliderThumbBoth.class
VerticalSliderThumbLeft.class          VerticalSliderThumbRight.class
WrappingLabel.class                    WrappingLabelBeanInfo.class
WrappingLabelC16.gif                   WrappingLabelC32.gif
[_vti_cnf]
            189 File(s)            652,892 bytes

Directory of D:\Inetpub\wwwroot\Survey\classes\symantec\itools\awt\image

[.]                     [..]                    DarkenFilter.class
FadeFilter.class        LightenFilter.class
              5 File(s)              7,631 bytes

Directory of D:\Inetpub\wwwroot\Survey\classes\symantec\itools\awt\multiList

[.]                                   [..]
Cell.class                            CompareTextAndImageCells.class
TextAndImageCell.class
              5 File(s)              4,490 bytes

Directory of D:\Inetpub\wwwroot\Survey\classes\symantec\itools\awt\shape

[.]                                   [..]
Circle.class                          CircleBeanInfo.class
CircleC16.gif                         CircleC32.gif
Ellipse.class                         EllipseBeanInfo.class
```

```
EllipseC16.gif                    EllipseC32.gif
HorizontalLine.class              HorizontalLineBeanInfo.class
HorizontalLineC16.gif             HorizontalLineC32.gif
Line.class                        LineBeanInfo.class
LineC16.gif                       LineC32.gif
Rect.class                        RectBeanInfo.class
RectC16.gif                       RectC32.gif
Shape.class                       ShapeBeanInfo.class
Square.class                      SquareBeanInfo.class
SquareC16.gif                     SquareC32.gif
VerticalLine.class                VerticalLineBeanInfo.class
VerticalLineC16.gif               VerticalLineC32.gif
[_vti_cnf]
             33 File(s)            47,004 bytes


  Directory of
D:\Inetpub\wwwroot\Survey\classes\symantec\itools\awt\shape\_vti_cnf

[.]                       [..]                      CircleC16.gif
CircleC32.gif             EllipseC16.gif            EllipseC32.gif
HorizontalLineC16.gif     HorizontalLineC32.gif     LineC16.gif
LineC32.gif               RectC16.gif               RectC32.gif
SquareC16.gif             SquareC32.gif             VerticalLineC16.gif
VerticalLineC32.gif
             16 File(s)            3,710 bytes


  Directory of D:\Inetpub\wwwroot\Survey\classes\symantec\itools\awt\util

[.]                               [..]
Calendar$Action.class             Calendar$Mouse.class
Calendar.class                    CalendarBeanInfo.class
CalendarC16.gif                   CalendarC32.gif
ColorUtils.class                  [dialog]
[edit]                            ProgressBar.class
ProgressBarBeanInfo.class         ProgressBarC16.gif
ProgressBarC32.gif                [spinner]
StatusScroller.class              StatusScrollerBeanInfo.class
StatusScrollerC16.gif             StatusScrollerC32.gif
ToolBarPanel$Veto.class           ToolBarPanel.class
ToolBarPanelBeanInfo.class        ToolBarPanelC16.gif
ToolBarPanelC32.gif               ToolBarSpacer$Veto.class
ToolBarSpacer.class               ToolBarSpacerBeanInfo.class
ToolBarSpacerC16.gif              ToolBarSpacerC32.gif
Util.class                        [_vti_cnf]
             32 File(s)            93,195 bytes


  Directory of D:\Inetpub\wwwroot\Survey\classes\symantec\itools\awt\util\dialog

[.]                  [..]              DialogBox.class     ModalDialog.class
              4 File(s)        2,449 bytes


  Directory of D:\Inetpub\wwwroot\Survey\classes\symantec\itools\awt\util\edit

[.]                               [..]
AddressCode.class                 AddressCodeBeanInfo.class
IntlLongDistPhoneNumber.class     IntlLongDistPhoneNumberBeanInfo.class
IntlLongDistPhoneNumberC16.gif    IntlLongDistPhoneNumberC32.gif
```

```
LocalPhoneNumber.class                  LocalPhoneNumberBeanInfo.class
LocalPhoneNumberC16.gif                 LocalPhoneNumberC32.gif
LongDistPhoneNumber.class               LongDistPhoneNumberBeanInfo.class
LongZipCode.class                       LongZipCodeBeanInfo.class
LongZipCodeC16.gif                      LongZipCodeC32.gif
PhoneNumber.class                       PhoneNumberBeanInfo.class
PostalCode.class                        PostalCodeBeanInfo.class
PostalCodeC16.gif                       PostalCodeC32.gif
SocialIDNumber.class                    SocialIDNumberBeanInfo.class
SocialInsuranceNumber.class             SocialInsuranceNumberBeanInfo.class
SocialInsuranceNumberC16.gif            SocialInsuranceNumberC32.gif
SocialSecurityNumber.class              SocialSecurityNumberBeanInfo.class
SocialSecurityNumberC16.gif             SocialSecurityNumberC32.gif
USLongDistPhoneNumber.class             USLongDistPhoneNumberBeanInfo.class
USLongDistPhoneNumberC16.gif            USLongDistPhoneNumberC32.gif
ZipCode.class                           ZipCodeBeanInfo.class
ZipCodeC16.gif                          ZipCodeC32.gif
[_vti_cnf]
                43 File(s)          44,980 bytes

  Directory of
D:\Inetpub\wwwroot\Survey\classes\symantec\itools\awt\util\edit\_vti_cnf

[.]                                     [..]
IntlLongDistPhoneNumberC16.gif          IntlLongDistPhoneNumberC32.gif
LocalPhoneNumberC16.gif                 LocalPhoneNumberC32.gif
LongZipCodeC16.gif                      LongZipCodeC32.gif
PostalCodeC16.gif                       PostalCodeC32.gif
SocialInsuranceNumberC16.gif            SocialInsuranceNumberC32.gif
SocialSecurityNumberC16.gif             SocialSecurityNumberC32.gif
USLongDistPhoneNumberC16.gif            USLongDistPhoneNumberC32.gif
ZipCodeC16.gif                          ZipCodeC32.gif
                18 File(s)           4,240 bytes

  Directory of D:\Inetpub\wwwroot\Survey\classes\symantec\itools\awt\util\spinner

[.]                                     [..]
DaySpinner.class                        DaySpinnerBeanInfo.class
DaySpinnerC16.gif                       DaySpinnerC32.gif
HorizontalSpinButtonPanel.class         ListSpinner.class
ListSpinnerBeanInfo.class               ListSpinnerC16.gif
ListSpinnerC32.gif                      MonthSpinner.class
MonthSpinnerBeanInfo.class              MonthSpinnerC16.gif
MonthSpinnerC32.gif                     NumericSpinner.class
NumericSpinnerBeanInfo.class            NumericSpinnerC16.gif
NumericSpinnerC32.gif                   SpinButtonPanel$Action.class
SpinButtonPanel.class                   Spinner$Action.class
Spinner$CurrentVeto.class               Spinner$MaxVeto.class
Spinner$MinVeto.class                   Spinner.class
SpinnerBeanInfo.class                   VerticalSpinButtonPanel.class
[_vti_cnf]
                29 File(s)          74,491 bytes

  Directory of
D:\Inetpub\wwwroot\Survey\classes\symantec\itools\awt\util\spinner\_vti_cnf

[.]                     [..]                    DaySpinnerC16.gif
```

290

```
DaySpinnerC32.gif        ListSpinnerC16.gif       ListSpinnerC32.gif
MonthSpinnerC16.gif      MonthSpinnerC32.gif      NumericSpinnerC16.gif
NumericSpinnerC32.gif
                10 File(s)           2,120 bytes


  Directory of
D:\Inetpub\wwwroot\Survey\classes\symantec\itools\awt\util\_vti_cnf

[.]                      [..]                     CalendarC16.gif
CalendarC32.gif          ProgressBarC16.gif       ProgressBarC32.gif
StatusScrollerC16.gif    StatusScrollerC32.gif    ToolBarPanelC16.gif
ToolBarPanelC32.gif      ToolBarSpacerC16.gif     ToolBarSpacerC32.gif
                12 File(s)           2,650 bytes


  Directory of D:\Inetpub\wwwroot\Survey\classes\symantec\itools\awt\_vti_cnf

[.]                           [..]
BorderPanelC16.gif            BorderPanelC32.gif
ComboBoxC16.gif               ComboBoxC32.gif
DirectionButtonC16.gif        DirectionButtonC32.gif
FormattedTextFieldC16.gif     FormattedTextFieldC32.gif
HorizontalSliderC16.gif       HorizontalSliderC32.gif
ImageButtonC16.gif            ImageButtonC32.gif
ImageHTMLLinkC16.gif          ImageHTMLLinkC32.gif
ImageListBoxC16.gif           ImageListBoxC32.gif
ImagePanelC16.gif             ImagePanelC32.gif
InvisibleButtonC16.gif        InvisibleButtonC32.gif
InvisibleHTMLLinkC16.gif      InvisibleHTMLLinkC32.gif
KeyPressManagerPanelC16.gif   KeyPressManagerPanelC32.gif
Label3DC16.gif                Label3DC32.gif
LabelButtonC16.gif            LabelButtonC32.gif
LabelHTMLLinkC16.gif          LabelHTMLLinkC32.gif
MultiListC16.gif              MultiListC32.gif
RadioButtonGroupPanelC16.gif  RadioButtonGroupPanelC32.gif
RollOverButtonC16.gif         RollOverButtonC32.gif
ScrollingPanelC16.gif         ScrollingPanelC32.gif
SplitterPanelC16.gif          SplitterPanelC32.gif
StateCheckBoxC16.gif          StateCheckBoxC32.gif
StatusBarC16.gif              StatusBarC32.gif
TabPanelC16.gif               TabPanelC32.gif
TreeViewC16.gif               TreeViewC32.gif
VerticalSliderC16.gif         VerticalSliderC32.gif
WrappingLabelC16.gif          WrappingLabelC32.gif
                54 File(s)         13,782 bytes


  Directory of D:\Inetpub\wwwroot\Survey\classes\symantec\itools\beans

[.]                           [..]
ConnectionDescriptor.class    PropertyChangeSupport.class
SymantecBeanDescriptor.class  VetoableChangeSupport.class
                6 File(s)          9,948 bytes


  Directory of D:\Inetpub\wwwroot\Survey\classes\symantec\itools\db

[.]                           [..]
[awt]                         ConnectionBeanIconColor16.gif
ConnectionBeanIconColor32.gif  DataSourceIconColor16.gif
```

```
DataSourceIconColor32.gif          MediatorDSIconColor16.gif
MediatorDSIconColor32.gif          MediatorIconColor16.gif
MediatorIconColor32.gif            [_vti_cnf]
              12 File(s)             4,624 bytes


 Directory of D:\Inetpub\wwwroot\Survey\classes\symantec\itools\db\awt

[.]                                [..]
ComboBoxC16.gif                    ComboBoxC32.gif
DBTstampC16.gif                    DBTstampC32.gif
FormattedTextFieldC16.gif          FormattedTextFieldC32.gif
NervousTextC16.gif                 NervousTextC32.gif
StateCheckBoxC16.gif               StateCheckBoxC32.gif
[_vti_cnf]
              13 File(s)             2,185 bytes


 Directory of D:\Inetpub\wwwroot\Survey\classes\symantec\itools\db\awt\_vti_cnf

[.]                                [..]
ComboBoxC16.gif                    ComboBoxC32.gif
DBTstampC16.gif                    DBTstampC32.gif
FormattedTextFieldC16.gif          FormattedTextFieldC32.gif
NervousTextC16.gif                 NervousTextC32.gif
StateCheckBoxC16.gif               StateCheckBoxC32.gif
              12 File(s)             2,650 bytes


 Directory of D:\Inetpub\wwwroot\Survey\classes\symantec\itools\db\_vti_cnf

[.]                                [..]
ConnectionBeanIconColor16.gif      ConnectionBeanIconColor32.gif
DataSourceIconColor16.gif          DataSourceIconColor32.gif
MediatorDSIconColor16.gif          MediatorDSIconColor32.gif
MediatorIconColor16.gif            MediatorIconColor32.gif
              10 File(s)             2,120 bytes


 Directory of D:\Inetpub\wwwroot\Survey\classes\symantec\itools\demo

[.]                 [..]                 Demo.class          DemoFrame.class
              4 File(s)             5,268 bytes


 Directory of D:\Inetpub\wwwroot\Survey\classes\symantec\itools\lang

[.]                 [..]                 Context.class    OS.class
              4 File(s)             3,962 bytes


 Directory of D:\Inetpub\wwwroot\Survey\classes\symantec\itools\multimedia

[.]                                [..]
Animator.class                     AnimatorBeanInfo.class
AnimatorC16.gif                    AnimatorC32.gif
AnimatorImage.class                Firework.class
FireworkBeanInfo.class             FireworkC16.gif
FireworkC32.gif                    ImageViewer.class
ImageViewerBeanInfo.class          ImageViewerC16.gif
ImageViewerC32.gif                 MovingAnimation.class
MovingAnimationBeanInfo.class      MovingAnimationC16.gif
MovingAnimationC32.gif             NervousText.class
```

```
NervousTextBeanInfo.class        NervousTextC16.gif
NervousTextC32.gif               Plasma.class
PlasmaBeanInfo.class             PlasmaC16.gif
PlasmaC32.gif                    Rocket.class
ScrollingText$Mouse.class        ScrollingText$MouseMotion.class
ScrollingText.class              ScrollingTextBeanInfo.class
ScrollingTextC16.gif             ScrollingTextC32.gif
SlideShow.class                  SoundPlayer.class
SoundPlayerBeanInfo.class        SoundPlayerC16.gif
SoundPlayerC32.gif               SoundViewerItem.class
SoundViewerThread.class          [_vti_cnf]
             42 File(s)       124,452 bytes


 Directory of
D:\Inetpub\wwwroot\Survey\classes\symantec\itools\multimedia\_vti_cnf

[.]                    [..]                    AnimatorC16.gif
AnimatorC32.gif        FireworkC16.gif         FireworkC32.gif
ImageViewerC16.gif     ImageViewerC32.gif      MovingAnimationC16.gif
MovingAnimationC32.gif NervousTextC16.gif      NervousTextC32.gif
PlasmaC16.gif          PlasmaC32.gif           ScrollingTextC16.gif
ScrollingTextC32.gif   SoundPlayerC16.gif      SoundPlayerC32.gif
             18 File(s)        4,240 bytes


 Directory of D:\Inetpub\wwwroot\Survey\classes\symantec\itools\net

[.]                    [..]                    RelativeURL.class
             3 File(s)        1,619 bytes


 Directory of D:\Inetpub\wwwroot\Survey\classes\symantec\itools\util

[.]                    [..]                    GeneralUtils.class
Timer.class            ZOrderUtils.class
             5 File(s)        9,479 bytes


 Directory of D:\Inetpub\wwwroot\Survey\classes\symantec\itools\vcafe

[.]                              [..]
[beanhelp]                       Bundler$Action.class
Bundler$Item.class               Bundler$Key.class
Bundler$SymText.class            Bundler$Window.class
Bundler.class                    BundlerNode.class
[commtool]                       DependentDialog$Action.class
DependentDialog$Key.class        DependentDialog$Window.class
DependentDialog.class            DependentItem.class
ErsatzFile.class                 InternalMapper.class
             18 File(s)        37,755 bytes


 Directory of D:\Inetpub\wwwroot\Survey\classes\symantec\itools\vcafe\beanhelp

[.]                              [..]
BackRunContainer.class           BackRunMemberInfo.class
BackRunMenu.class                BackRunMenuBar.class
BackrunObjectCreator.class       BackRunParent.class
[beaninfo]                       ClassInfo.class
ConvertToVPO$Helper.class        ConvertToVPO.class
DoOnBean.class                   [editors]
```

```
ImageData.class                      InputStreamProducer.class
JarClassInfo.class                   JarInfo.class
JarManager.class                     LoadedJar.class
Manifest.class                       MessageHeader.class
ObjectInfo.class                     ObjInputStreamWithLoader.class
SimpleClassLoader.class              [simpleresource]
            26 File(s)               77,551 bytes


 Directory of
D:\Inetpub\wwwroot\Survey\classes\symantec\itools\vcafe\beanhelp\beaninfo

[.]                                  [..]
CanvasBeanInfo.class                 ComponentBeanInfo.class
ContainerBeanInfo.class              PanelBeanInfo.class
TextComponentBeanInfo.class          TextFieldBeanInfo.class
             8 File(s)               24,595 bytes


 Directory of
D:\Inetpub\wwwroot\Survey\classes\symantec\itools\vcafe\beanhelp\editors

[.]                                  [..]
BooleanEditor.class                  CharacterEditor.class
CheckboxGroupEditor.class            ColorEditor.class
CursorEditor.class                   FontEditor.class
IntegerEditor.class                  LayoutManagerEditor.class
RectangleEditor.class                StringArrayEditor.class
StringEditor.class                   SymPropertyEditorSupport.class
URLArrayEditor.class                 URLEditor.class
UseParentsBackgroundEditor.class     UseParentsFontEditor.class
UseParentsForegroundEditor.class
            19 File(s)               18,547 bytes


 Directory of
D:\Inetpub\wwwroot\Survey\classes\symantec\itools\vcafe\beanhelp\simpleresource

[.]                                  [..]
Handler.class                        SimpleResourceConnection.class
             4 File(s)               3,042 bytes


 Directory of D:\Inetpub\wwwroot\Survey\classes\symantec\itools\vcafe\commtool

[.]                                  [..]
Memory.class                         MeterTool$1.class
MeterTool$SymItem.class              MeterTool$SymWindow.class
MeterTool.class
             7 File(s)               7,771 bytes


 Directory of D:\Inetpub\wwwroot\Survey\classes\symantec\tools

[.]          [..]        [debug]     [java]     [javac]     [ttydebug]
             6 File(s)               0 bytes


 Directory of D:\Inetpub\wwwroot\Survey\classes\symantec\tools\debug

[.]
[..]
Agent.class
```

```
AgentConstants.class
AgentIn.class
AgentOutputStream.class
BreakpointHandler.class
BreakpointQueue.class
BreakpointSet.class
DebuggerCallback.class
EmptyApp.class
ExecReRouter.class
ExpressionOutOfContextException.class
ExpressionWrongNumberLocalVarsException.class
Field.class
InvalidPCException.class
LineNumber.class
LocalCommInputStream.class
LocalCommOutputStream.class
LocalCommStream.class
LocalVariable.class
MainThread.class
NoSessionException.class
NoSuchFieldException.class
NoSuchLineNumberException.class
RemoteAgent.class
RemoteArray.class
RemoteBoolean.class
RemoteByte.class
RemoteChar.class
RemoteClass.class
RemoteDebugger.class
RemoteDouble.class
RemoteField.class
RemoteFloat.class
RemoteInt.class
RemoteLong.class
RemoteObject.class
RemoteShort.class
RemoteStackFrame.class
RemoteStackVariable.class
RemoteString.class
RemoteThread.class
RemoteThreadGroup.class
RemoteValue.class
ResponseStream.class
StackFrame.class
SymDbgExec.class
SymDbgExprFindContext.class
SymDbgExprHandler.class
SymDbgExprResult.class
SymDbgLoadExpr.class
SymDbgNotify.class
SymDbgProcess.class
ThreadList.class
            55 File(s)        136,888 bytes
```

Directory of D:\Inetpub\wwwroot\Survey\classes\symantec\tools\java

[.]   [..]

```
VEDbgJavaDataRec.class          VEDbgJavaDebug.class
VEDbgJavaIntf.class             VEDbgJavaKernel.class
VEDbgJavaMsgRouter.class        VEDbgJavaRemoteApp.class
VEDbgJavaResource.class         VEDbgJavaResource_ja.class
VEDbgJavaRuntimeCmd.class
                11 File(s)          39,759 bytes


 Directory of D:\Inetpub\wwwroot\Survey\classes\symantec\tools\javac

[.]                      [..]                    BatchEnvironment.class
BatchParser.class        ErrorConsumer.class     ErrorMessage.class
JavaIDDE.class           Main.class
                8 File(s)           33,103 bytes


 Directory of D:\Inetpub\wwwroot\Survey\classes\symantec\tools\ttydebug

[.]            [..]           TTY.class
                3 File(s)           28,429 bytes


 Directory of D:\Inetpub\wwwroot\Survey\classes\weblogic

[.]      [..]     [jdbc]
                3 File(s)                  0 bytes


 Directory of D:\Inetpub\wwwroot\Survey\classes\weblogic\jdbc

[.]              [..]             [common4]    [microsoft4] [sybase4]
                5 File(s)                  0 bytes


 Directory of D:\Inetpub\wwwroot\Survey\classes\weblogic\jdbc\common4

[.]              [..]             [compression]  [cryptography] [io]
[net]            [sql]            [util]
                8 File(s)                  0 bytes


 Directory of
D:\Inetpub\wwwroot\Survey\classes\weblogic\jdbc\common4\compression

[.]                      [..]                    Compressor.class
LzssCompressor.class     LzssNode.class          TdsCompressor.class
                6 File(s)           3,880 bytes


 Directory of
D:\Inetpub\wwwroot\Survey\classes\weblogic\jdbc\common4\cryptography

[.]                      [..]
BlowfishCryptographer.class     Cryptographer.class
DESCryptographer.class          XorCryptographer.class
                6 File(s)           38,039 bytes


 Directory of D:\Inetpub\wwwroot\Survey\classes\weblogic\jdbc\common4\io

[.]                      [..]
AsciiInputStream.class           BufferedFilterInputStream.class
BufferedFilterOutputStream.class CompressedInputStream.class
CompressedOutputStream.class     EncryptedInputStream.class
EncryptedOutputStream.class      HexInputStream.class
```

UnicodeInputStream.class
                11 File(s)   10,326 bytes

 Directory of D:\Inetpub\wwwroot\Survey\classes\weblogic\jdbc\common4\net

[.]   [..]   SocketExtras.class
SocketTimeout.class
                4 File(s)   4,352 bytes

 Directory of D:\Inetpub\wwwroot\Survey\classes\weblogic\jdbc\common4\sql

[.]   [..]
AsyncWriter.class   BaseConnection.class
BaseHandler.class   BaseString.class
ConnectDriver.class   ConnectionInfo.class
JdbcExpandedSqlParser.class   MicrosoftConnection.class
MicrosoftDatabaseMetaData.class   MicrosoftLoginStatement.class
TdsCallableStatement.class   TdsColumnMetaData.class
TdsConnection.class   TdsDatabaseMetaData.class
TdsEntry.class   TdsException.class
TdsInputStream.class   TdsLoginStatement.class
TdsOutputStream.class   TdsParameter.class
TdsPreparedInputStream.class   TdsPreparedStatement.class
TdsResultSet.class   TdsResultSetMetaData.class
TdsStatement.class
                27 File(s)   142,174 bytes

 Directory of D:\Inetpub\wwwroot\Survey\classes\weblogic\jdbc\common4\util

[.]   [..]   BitsArray.class   Day.class
                4 File(s)   2,822 bytes

 Directory of D:\Inetpub\wwwroot\Survey\classes\weblogic\jdbc\microsoft4

[.]   [..]   Driver.class
                3 File(s)   859 bytes

 Directory of D:\Inetpub\wwwroot\Survey\classes\weblogic\jdbc\sybase4

[.]   [..]   Driver.class
                3 File(s)   841 bytes

 Directory of D:\Inetpub\wwwroot\Survey\images

[.]   [..]   COMPASS.GIF   Connie.jpg   DENIM.JPG
DESERT.GIF   FLY.GIF   help.gif   IND-DECK.GIF   IND-STRN.GIF
RADAR.GIF   [_vti_cnf]
                12 File(s)   240,069 bytes

 Directory of D:\Inetpub\wwwroot\Survey\images_vti_cnf

[.]   [..]   COMPASS.GIF   Connie.jpg   DENIM.JPG
DESERT.GIF   FLY.GIF   IND-DECK.GIF   IND-STRN.GIF   RADAR.GIF
                10 File(s)   2,253 bytes

 Directory of D:\Inetpub\wwwroot\Survey\javascript

```
[.]                   [..]                      checksurvey.js    clearsurvey.js
[_vti_cnf]
                      5 File(s)              9,549 bytes

   Directory of D:\Inetpub\wwwroot\Survey\javascript\_vti_cnf

[.]                   [..]                      checksurvey.js
                      3 File(s)               117 bytes

   Directory of D:\Inetpub\wwwroot\Survey\sounds

[.]                   [..]               bluesbro.mid   GONNAFLY.MID    mission.mid
SAWYER.MID        viewkill.mid
                      7 File(s)            196,659 bytes

   Directory of D:\Inetpub\wwwroot\Survey\_vti_cnf

[.]                   [..]               Applet.html    Survey.html
                      4 File(s)            1,156 bytes

   Directory of D:\Inetpub\wwwroot\surveyadmin

[.]                   [..]               ADOVBS.inc    global.asa    test.asp
                      5 File(s)            9,708 bytes

   Directory of D:\Inetpub\wwwroot\time

[.]                   [..]               [classes]        TimeZone.html
                      4 File(s)            1,074 bytes

   Directory of D:\Inetpub\wwwroot\time\classes

[.]
[..]
AboutDialog$SymAction.class
AboutDialog$SymWindow.class
AboutDialog.class
AboutDialog.java
ColorPalette$SymMouse.class
ColorPalette$SymMouse.obj
ColorPalette$SymMouseMotion.class
ColorPalette$SymMouseMotion.obj
ColorPalette$SymWindow.class
ColorPalette.class
ColorPalette.java
QuitDialog$SymAction.class
QuitDialog$SymWindow.class
QuitDialog.class
QuitDialog.java
[symantec]
TimeApplet$SymMouse.class
TimeApplet.class
TimeApplet.java
TimeFrameApplication$SymAction.class
TimeFrameApplication$SymComponent.class
TimeFrameApplication$SymMouse.class
TimeFrameApplication$SymWindow.class
```

```
TimeFrameApplication.class
TimeFrameApplication.java
TimeZone.html
TimeZoneUpdater.class
TimeZoneUpdater.java
              30 File(s)           75,143 bytes

 Directory of D:\Inetpub\wwwroot\time\classes\symantec

[.]        [..]       [beans]  [dti]    [itools] [tools]
              6 File(s)              0 bytes

 Directory of D:\Inetpub\wwwroot\time\classes\symantec\beans

[.]                 [..]          Beans.class
              3 File(s)            734 bytes

 Directory of D:\Inetpub\wwwroot\time\classes\symantec\dti

[.]
[..]
BogusComponent.class
ComboBoxDesignTimeInterface.class
ControlDesignTimeInterface.class
CustomPropEditFrame.class
DefaultAwtComponentDesignTimeInterface.class
DefaultAwtPanelDesignTimeInterface.class
EventInfo.class
EventParamInfo.class
JBlenderFrame.class
JBlenderFrameCreateParams.class
JBlenderMain.class
JBlenderPanel.class
JBlenderPanelGridInfo.class
JBlenderReqQueue.class
JBlenderRequest.class
JClassLoader.class
JCustomPropFrame.class
JDebugStr.class
JInvokeThread.class
NewFont.class
PropInfo.class
PropInfoBoolean.class
PropInfoCharacter.class
PropInfoColor.class
PropInfoCustom.class
PropInfoDouble.class
PropInfoDynamic.class
PropInfoEnum.class
PropInfoFont.class
PropInfoInteger.class
PropInfoInterface.class
PropInfoString.class
PropInfoStringArray.class
PropInfoURL.class
PropMethod.class
Queue.class
```

```
ScrollingNames$JPActionHandler.class
ScrollingNames$SymComponent.class
ScrollingNames.class
ScrollingPanelDesignTimeInterface.class
SlideShowDesignTimeInterface.class
StarMain$SymWindow.class
StarMain.class
stars.gif
StdAwtButtonDesignTimeInterface.class
StdAwtCheckboxDesignTimeInterface.class
StdAwtChoiceDesignTimeInterface.class
StdAwtLabelDesignTimeInterface.class
StdAwtScrollbarDesignTimeInterface.class
StdAwtTextAreaDesignTimeInterface.class
StdAwtTextComponentDesignTimeInterface.class
StdAwtTextFieldDesignTimeInterface.class
symName.class
TabPanelDesignTimeInterface.class
TestFrame.class
TreeViewDesignTimeInterface.class
VCFormCompListener$AddRemoveListener.class
VCFormCompListener$Item.class
VCFormCompListener.class
WrappingLabelDesignTimeInterface.class
[_vti_cnf]
              63 File(s)          111,563 bytes


 Directory of D:\Inetpub\wwwroot\time\classes\symantec\dti\_vti_cnf

[.]            [..]           stars.gif
                  3 File(s)              265 bytes


 Directory of D:\Inetpub\wwwroot\time\classes\symantec\itools

[.]            [..]         [awt]        [beans]       [db]          [demo]
[lang]         [multimedia] [net]        [util]        [vcafe]
                 11 File(s)            0 bytes


 Directory of D:\Inetpub\wwwroot\time\classes\symantec\itools\awt

[.]                                      [..]
AlignStyle.class                         BaseTabbedPanel$Action.class
BaseTabbedPanel$Mouse.class              BaseTabbedPanel.class
BaseTabbedPanelBeanInfo.class            BevelStyle.class
BorderPanel$Veto.class                   BorderPanel.class
BorderPanelBeanInfo.class                BorderPanelC16.gif
BorderPanelC32.gif                       ButtonBase$Action.class
ButtonBase$BevelVeto.class               ButtonBase$Focus.class
ButtonBase$FrameVeto.class               ButtonBase$Key.class
ButtonBase$Mouse.class                   ButtonBase.class
ButtonBaseBeanInfo.class                 ComboBox$Action.class
ComboBox$Focus.class                     ComboBox$Item.class
ComboBox$KeyBox.class                    ComboBox$KeyList.class
ComboBox$Mouse.class                     ComboBox.class
ComboBoxBeanInfo.class                   ComboBoxC16.gif
ComboBoxC32.gif                          CompareCells.class
CompareFunc.class                        CompareFuncCB.class
```

```
DirectionButton$IndntVeto.class          DirectionButton$SizeVeto.class
DirectionButton.class                    DirectionButtonBeanInfo.class
DirectionButtonC16.gif                   DirectionButtonC32.gif
FormattedTextField$Key.class             FormattedTextField.class
FormattedTextFieldBeanInfo.class         FormattedTextFieldC16.gif
FormattedTextFieldC32.gif                HorizontalSlider$Mouse.class
HorizontalSlider$MouseMtn.class          HorizontalSlider.class
HorizontalSliderBeanInfo.class           HorizontalSliderC16.gif
HorizontalSliderC32.gif                  HorizontalSliderThumb.class
HorizontalSliderThumbBot.class           HorizontalSliderThumbBoth.class
HorizontalSliderThumbTop.class           [image]
ImageButton$StyleVeto.class              ImageButton.class
ImageButtonBeanInfo.class                ImageButtonC16.gif
ImageButtonC32.gif                       ImageHTMLLink.class
ImageHTMLLinkBeanInfo.class              ImageHTMLLinkC16.gif
ImageHTMLLinkC32.gif                     ImageListBox$Adjustment.class
ImageListBox$Key.class                   ImageListBox$Mouse.class
ImageListBox$MouseMotion.class           ImageListBox.class
ImageListBoxBeanInfo.class               ImageListBoxC16.gif
ImageListBoxC32.gif                      ImagePanel.class
ImagePanelBeanInfo.class                 ImagePanelC16.gif
ImagePanelC32.gif                        InvalidTreeNodeException.class
InvisibleButton$Mouse.class              InvisibleButton.class
InvisibleButtonBeanInfo.class            InvisibleButtonC16.gif
InvisibleButtonC32.gif                   InvisibleHTMLLink$Action.class
InvisibleHTMLLink$Mouse.class            InvisibleHTMLLink.class
InvisibleHTMLLinkBeanInfo.class          InvisibleHTMLLinkC16.gif
InvisibleHTMLLinkC32.gif                 KeyPressManagerPanel$A.class
KeyPressManagerPanel$K.class             KeyPressManagerPanel.class
KeyPressManagerPanelBeanInfo.class       KeyPressManagerPanelC16.gif
KeyPressManagerPanelC32.gif              Label3D.class
Label3DBeanInfo.class                    Label3DC16.gif
Label3DC32.gif                           LabelButton$HAVeto.class
LabelButton$VAVeto.class                 LabelButton.class
LabelButtonBeanInfo.class                LabelButtonC16.gif
LabelButtonC32.gif                       LabelHTMLLink.class
LabelHTMLLinkBeanInfo.class              LabelHTMLLinkC16.gif
LabelHTMLLinkC32.gif                     ListItem.class
Matrix.class                             MatrixEnumeration.class
[multiList]                              MultiList$Adjustment.class
MultiList$Focus.class                    MultiList$Key.class
MultiList$Mouse.class                    MultiList$MouseMotion.class
MultiList.class                          MultiListBeanInfo.class
MultiListC16.gif                         MultiListC32.gif
Orientation.class                        RadioButtonGroupPanel.class
RadioButtonGroupPanelBeanInfo.class      RadioButtonGroupPanelC16.gif
RadioButtonGroupPanelC32.gif             RollOverButton$Mouse.class
RollOverButton$MouseMtn.class            RollOverButton.class
RollOverButtonBeanInfo.class             RollOverButtonC16.gif
RollOverButtonC32.gif                    ScrollingPanel.class
ScrollingPanelBeanInfo.class             ScrollingPanelC16.gif
ScrollingPanelC32.gif                    [shape]
Slider.class                             SliderBeanInfo.class
SliderTick.class                         SplitterPanel$Mouse.class
SplitterPanel$MouseMtn.class             SplitterPanel$STVeto.class
SplitterPanel.class                      SplitterPanelBeanInfo.class
SplitterPanelC16.gif                     SplitterPanelC32.gif
```

```
StateCheckBox$Mouse.class              StateCheckBox.class
StateCheckBoxBeanInfo.class            StateCheckBoxC16.gif
StateCheckBoxC32.gif                   StatusBar.class
StatusBarBeanInfo.class                StatusBarC16.gif
StatusBarC32.gif                       TabPanel$PropertyChange.class
TabPanel.class                         TabPanelBeanInfo.class
TabPanelC16.gif                        TabPanelC32.gif
TransparencyTrick.class                TransparencyTrickUtils.class
TreeNode.class                         TreeView$Adjustment.class
TreeView$Focus.class                   TreeView$Key.class
TreeView$Mouse.class                   TreeView.class
TreeViewBeanInfo.class                 TreeViewC16.gif
TreeViewC32.gif                        [util]
VerticalSlider$Mouse.class             VerticalSlider$MouseMtn.class
VerticalSlider.class                   VerticalSliderBeanInfo.class
VerticalSliderC16.gif                  VerticalSliderC32.gif
VerticalSliderThumb.class              VerticalSliderThumbBoth.class
VerticalSliderThumbLeft.class          VerticalSliderThumbRight.class
WrappingLabel.class                    WrappingLabelBeanInfo.class
WrappingLabelC16.gif                   WrappingLabelC32.gif
[_vti_cnf]
            189 File(s)          652,892 bytes

    Directory of D:\Inetpub\wwwroot\time\classes\symantec\itools\awt\image

[.]                     [..]                    DarkenFilter.class
FadeFilter.class        LightenFilter.class
            5 File(s)            7,631 bytes

    Directory of D:\Inetpub\wwwroot\time\classes\symantec\itools\awt\multiList

[.]                     [..]
Cell.class              CompareTextAndImageCells.class
TextAndImageCell.class
            5 File(s)            4,490 bytes

    Directory of D:\Inetpub\wwwroot\time\classes\symantec\itools\awt\shape

[.]                     [..]
Circle.class            CircleBeanInfo.class
CircleC16.gif           CircleC32.gif
Ellipse.class           EllipseBeanInfo.class
EllipseC16.gif          EllipseC32.gif
HorizontalLine.class    HorizontalLineBeanInfo.class
HorizontalLineC16.gif   HorizontalLineC32.gif
Line.class              LineBeanInfo.class
LineC16.gif             LineC32.gif
Rect.class              RectBeanInfo.class
RectC16.gif             RectC32.gif
Shape.class             ShapeBeanInfo.class
Square.class            SquareBeanInfo.class
SquareC16.gif           SquareC32.gif
VerticalLine.class      VerticalLineBeanInfo.class
VerticalLineC16.gif     VerticalLineC32.gif
[_vti_cnf]
            33 File(s)            47,004 bytes
```

```
Directory of D:\Inetpub\wwwroot\time\classes\symantec\itools\awt\shape\_vti_cnf

[.]                        [..]                       CircleC16.gif
CircleC32.gif              EllipseC16.gif             EllipseC32.gif
HorizontalLineC16.gif      HorizontalLineC32.gif      LineC16.gif
LineC32.gif                RectC16.gif                RectC32.gif
SquareC16.gif              SquareC32.gif              VerticalLineC16.gif
VerticalLineC32.gif
                16 File(s)            3,710 bytes


Directory of D:\Inetpub\wwwroot\time\classes\symantec\itools\awt\util

[.]                              [..]
Calendar$Action.class            Calendar$Mouse.class
Calendar.class                   CalendarBeanInfo.class
CalendarC16.gif                  CalendarC32.gif
ColorUtils.class                 [dialog]
[edit]                           ProgressBar.class
ProgressBarBeanInfo.class        ProgressBarC16.gif
ProgressBarC32.gif               [spinner]
StatusScroller.class             StatusScrollerBeanInfo.class
StatusScrollerC16.gif            StatusScrollerC32.gif
ToolBarPanel$Veto.class          ToolBarPanel.class
ToolBarPanelBeanInfo.class       ToolBarPanelC16.gif
ToolBarPanelC32.gif              ToolBarSpacer$Veto.class
ToolBarSpacer.class              ToolBarSpacerBeanInfo.class
ToolBarSpacerC16.gif             ToolBarSpacerC32.gif
Util.class                       [_vti_cnf]
                32 File(s)             93,195 bytes


Directory of D:\Inetpub\wwwroot\time\classes\symantec\itools\awt\util\dialog

[.]                [..]                 DialogBox.class      ModalDialog.class
                4 File(s)         2,449 bytes


Directory of D:\Inetpub\wwwroot\time\classes\symantec\itools\awt\util\edit

[.]                                [..]
AddressCode.class                  AddressCodeBeanInfo.class
IntlLongDistPhoneNumber.class      IntlLongDistPhoneNumberBeanInfo.class
IntlLongDistPhoneNumberC16.gif     IntlLongDistPhoneNumberC32.gif
LocalPhoneNumber.class             LocalPhoneNumberBeanInfo.class
LocalPhoneNumberC16.gif            LocalPhoneNumberC32.gif
LongDistPhoneNumber.class          LongDistPhoneNumberBeanInfo.class
LongZipCode.class                  LongZipCodeBeanInfo.class
LongZipCodeC16.gif                 LongZipCodeC32.gif
PhoneNumber.class                  PhoneNumberBeanInfo.class
PostalCode.class                   PostalCodeBeanInfo.class
PostalCodeC16.gif                  PostalCodeC32.gif
SocialIDNumber.class               SocialIDNumberBeanInfo.class
SocialInsuranceNumber.class        SocialInsuranceNumberBeanInfo.class
SocialInsuranceNumberC16.gif       SocialInsuranceNumberC32.gif
SocialSecurityNumber.class         SocialSecurityNumberBeanInfo.class
SocialSecurityNumberC16.gif        SocialSecurityNumberC32.gif
USLongDistPhoneNumber.class        USLongDistPhoneNumberBeanInfo.class
USLongDistPhoneNumberC16.gif       USLongDistPhoneNumberC32.gif
ZipCode.class                      ZipCodeBeanInfo.class
```

ZipCodeC16.gif   ZipCodeC32.gif
[_vti_cnf]
                43 File(s)   44,980 bytes

 Directory of
D:\Inetpub\wwwroot\time\classes\symantec\itools\awt\util\edit_vti_cnf

[.]   [..]
IntlLongDistPhoneNumberC16.gif   IntlLongDistPhoneNumberC32.gif
LocalPhoneNumberC16.gif   LocalPhoneNumberC32.gif
LongZipCodeC16.gif   LongZipCodeC32.gif
PostalCodeC16.gif   PostalCodeC32.gif
SocialInsuranceNumberC16.gif   SocialInsuranceNumberC32.gif
SocialSecurityNumberC16.gif   SocialSecurityNumberC32.gif
USLongDistPhoneNumberC16.gif   USLongDistPhoneNumberC32.gif
ZipCodeC16.gif   ZipCodeC32.gif
                18 File(s)   4,240 bytes

 Directory of D:\Inetpub\wwwroot\time\classes\symantec\itools\awt\util\spinner

[.]   [..]
DaySpinner.class   DaySpinnerBeanInfo.class
DaySpinnerC16.gif   DaySpinnerC32.gif
HorizontalSpinButtonPanel.class   ListSpinner.class
ListSpinnerBeanInfo.class   ListSpinnerC16.gif
ListSpinnerC32.gif   MonthSpinner.class
MonthSpinnerBeanInfo.class   MonthSpinnerC16.gif
MonthSpinnerC32.gif   NumericSpinner.class
NumericSpinnerBeanInfo.class   NumericSpinnerC16.gif
NumericSpinnerC32.gif   SpinButtonPanel$Action.class
SpinButtonPanel.class   Spinner$Action.class
Spinner$CurrentVeto.class   Spinner$MaxVeto.class
Spinner$MinVeto.class   Spinner.class
SpinnerBeanInfo.class   VerticalSpinButtonPanel.class
[_vti_cnf]
                29 File(s)   74,491 bytes

 Directory of
D:\Inetpub\wwwroot\time\classes\symantec\itools\awt\util\spinner_vti_cnf

[.]   [..]   DaySpinnerC16.gif
DaySpinnerC32.gif   ListSpinnerC16.gif   ListSpinnerC32.gif
MonthSpinnerC16.gif   MonthSpinnerC32.gif   NumericSpinnerC16.gif
NumericSpinnerC32.gif
                10 File(s)   2,120 bytes

 Directory of D:\Inetpub\wwwroot\time\classes\symantec\itools\awt\util_vti_cnf

[.]   [..]   CalendarC16.gif
CalendarC32.gif   ProgressBarC16.gif   ProgressBarC32.gif
StatusScrollerC16.gif   StatusScrollerC32.gif   ToolBarPanelC16.gif
ToolBarPanelC32.gif   ToolBarSpacerC16.gif   ToolBarSpacerC32.gif
                12 File(s)   2,650 bytes

 Directory of D:\Inetpub\wwwroot\time\classes\symantec\itools\awt_vti_cnf

[.]   [..]

```
BorderPanelC16.gif                  BorderPanelC32.gif
ComboBoxC16.gif                     ComboBoxC32.gif
DirectionButtonC16.gif              DirectionButtonC32.gif
FormattedTextFieldC16.gif           FormattedTextFieldC32.gif
HorizontalSliderC16.gif             HorizontalSliderC32.gif
ImageButtonC16.gif                  ImageButtonC32.gif
ImageHTMLLinkC16.gif                ImageHTMLLinkC32.gif
ImageListBoxC16.gif                 ImageListBoxC32.gif
ImagePanelC16.gif                   ImagePanelC32.gif
InvisibleButtonC16.gif              InvisibleButtonC32.gif
InvisibleHTMLLinkC16.gif            InvisibleHTMLLinkC32.gif
KeyPressManagerPanelC16.gif         KeyPressManagerPanelC32.gif
Label3DC16.gif                      Label3DC32.gif
LabelButtonC16.gif                  LabelButtonC32.gif
LabelHTMLLinkC16.gif                LabelHTMLLinkC32.gif
MultiListC16.gif                    MultiListC32.gif
RadioButtonGroupPanelC16.gif        RadioButtonGroupPanelC32.gif
RollOverButtonC16.gif               RollOverButtonC32.gif
ScrollingPanelC16.gif               ScrollingPanelC32.gif
SplitterPanelC16.gif                SplitterPanelC32.gif
StateCheckBoxC16.gif                StateCheckBoxC32.gif
StatusBarC16.gif                    StatusBarC32.gif
TabPanelC16.gif                     TabPanelC32.gif
TreeViewC16.gif                     TreeViewC32.gif
VerticalSliderC16.gif               VerticalSliderC32.gif
WrappingLabelC16.gif                WrappingLabelC32.gif
            54 File(s)         13,782 bytes
```

Directory of D:\Inetpub\wwwroot\time\classes\symantec\itools\beans

```
[.]                                 [..]
ConnectionDescriptor.class          PropertyChangeSupport.class
SymantecBeanDescriptor.class        VetoableChangeSupport.class
             6 File(s)          9,948 bytes
```

Directory of D:\Inetpub\wwwroot\time\classes\symantec\itools\db

```
[.]                                 [..]
[awt]                               ConnectionBeanIconColor16.gif
ConnectionBeanIconColor32.gif       DataSourceIconColor16.gif
DataSourceIconColor32.gif           MediatorDSIconColor16.gif
MediatorDSIconColor32.gif           MediatorIconColor16.gif
MediatorIconColor32.gif             [_vti_cnf]
            12 File(s)          4,624 bytes
```

Directory of D:\Inetpub\wwwroot\time\classes\symantec\itools\db\awt

```
[.]                                 [..]
ComboBoxC16.gif                     ComboBoxC32.gif
DBTstampC16.gif                     DBTstampC32.gif
FormattedTextFieldC16.gif           FormattedTextFieldC32.gif
NervousTextC16.gif                  NervousTextC32.gif
StateCheckBoxC16.gif                StateCheckBoxC32.gif
[_vti_cnf]
            13 File(s)          2,185 bytes
```

Directory of D:\Inetpub\wwwroot\time\classes\symantec\itools\db\awt_vti_cnf

305

```
[.]         '               [..]
ComboBoxC16.gif             ComboBoxC32.gif
DBTstampC16.gif             DBTstampC32.gif
FormattedTextFieldC16.gif   FormattedTextFieldC32.gif
NervousTextC16.gif          NervousTextC32.gif
StateCheckBoxC16.gif        StateCheckBoxC32.gif
           12 File(s)          2,650 bytes


   Directory of D:\Inetpub\wwwroot\time\classes\symantec\itools\db\_vti_cnf


[.]                         [..]
ConnectionBeanIconColor16.gif   ConnectionBeanIconColor32.gif
DataSourceIconColor16.gif       DataSourceIconColor32.gif
MediatorDSIconColor16.gif       MediatorDSIconColor32.gif
MediatorIconColor16.gif         MediatorIconColor32.gif
           10 File(s)          2,120 bytes   ·


   Directory of D:\Inetpub\wwwroot\time\classes\symantec\itools\demo


[.]               [..]              Demo.class         DemoFrame.class
              ·  4 File(s)      5,268 bytes


   Directory of D:\Inetpub\wwwroot\time\classes\symantec\itools\lang


[.]               [..]              Context.class   ' OS.class
                 4 File(s)      3,962 bytes


   Directory of D:\Inetpub\wwwroot\time\classes\symantec\itools\multimedia


[.]                   .             [..]
Animator.class                      AnimatorBeanInfo.class
AnimatorC16.gif               .     AnimatorC32.gif      ·
AnimatorImage.class                 Firework.class
FireworkBeanInfo.class              FireworkC16.gif
FireworkC32.gif                     ImageViewer.class
ImageViewerBeanInfo.class           ImageViewerC16.gif
ImageViewerC32.gif                  MovingAnimation.class
MovingAnimationBeanInfo.class       MovingAnimationC16.gif
MovingAnimationC32.gif              NervousText.class
NervousTextBeanInfo.class           NervousTextC16.gif
NervousTextC32.gif                  Plasma.class
PlasmaBeanInfo.class                PlasmaC16.gif
PlasmaC32.gif                       Rocket.class
ScrollingText$Mouse.class           ScrollingText$MouseMotion.class
ScrollingText.class                 ScrollingTextBeanInfo.class
ScrollingTextC16.gif                ScrollingTextC32.gif
SlideShow.class                     SoundPlayer.class
SoundPlayerBeanInfo.class           SoundPlayerC16.gif
SoundPlayerC32.gif                  SoundViewerItem.class
SoundViewerThread.class             [_vti_cnf]
           42 File(s)        124,452 bytes


 Directory of
D:\Inetpub\wwwroot\time\classes\symantec\itools\multimedia\_vti_cnf


[.]                   [..]                      AnimatorC16.gif
```

306

```
AnimatorC32.gif          FireworkC16.gif          FireworkC32.gif
ImageViewerC16.gif       ImageViewerC32.gif       MovingAnimationC16.gif
MovingAnimationC32.gif   NervousTextC16.gif       NervousTextC32.gif
PlasmaC16.gif            PlasmaC32.gif            ScrollingTextC16.gif
ScrollingTextC32.gif     SoundPlayerC16.gif       SoundPlayerC32.gif
            18 File(s)           4,240 bytes


 Directory of D:\Inetpub\wwwroot\time\classes\symantec\itools\net

[.]                    [..]                    RelativeURL.class
            3 File(s)           1,619 bytes


 Directory of D:\Inetpub\wwwroot\time\classes\symantec\itools\util

[.]                    [..]                       GeneralUtils.class
Timer.class            ZOrderUtils.class
            5 File(s)           9,479 bytes


 Directory of D:\Inetpub\wwwroot\time\classes\symantec\itools\vcafe

[.]                             [..]
[beanhelp]                      Bundler$Action.class
Bundler$Item.class              Bundler$Key.class
Bundler$SymText.class           Bundler$Window.class
Bundler.class                   BundlerNode.class
[commtool]                      DependentDialog$Action.class
DependentDialog$Key.class       DependentDialog$Window.class
DependentDialog.class           DependentItem.class
ErsatzFile.class                InternalMapper.class
            18 File(s)       .   37,755 bytes


 Directory of D:\Inetpub\wwwroot\time\classes\symantec\itools\vcafe\beanhelp

[.]               .             [..]
BackRunContainer.class          BackRunMemberInfo.class
BackRunMenu.class               BackRunMenuBar.class
BackrunObjectCreator.class      BackRunParent.class
[beaninfo]                      ClassInfo.class
ConvertToVPO$Helper.class       ConvertToVPO.class
DoOnBean.class                  [editors]
ImageData.class                 InputStreamProducer.class
JarClassInfo.class              JarInfo.class
JarManager.class                LoadedJar.class
Manifest.class                  MessageHeader.class
ObjectInfo.class                ObjInputStreamWithLoader.class
SimpleClassLoader.class .       [simpleresource]
            26 File(s)           77,551 bytes


 Directory of
D:\Inetpub\wwwroot\time\classes\symantec\itools\vcafe\beanhelp\beaninfo

[.]                             [..]     /
CanvasBeanInfo.class            ComponentBeanInfo.class
ContainerBeanInfo.class         PanelBeanInfo.class
TextComponentBeanInfo.class     TextFieldBeanInfo.class
            8 File(s)           24,595 bytes
```

```
   Directory of
D:\Inetpub\wwwroot\time\classes\symantec\itools\vcafe\beanhelp\editors

[.]                              [..]
BooleanEditor.class              CharacterEditor.class
CheckboxGroupEditor.class        ColorEditor.class
CursorEditor.class               FontEditor.class
IntegerEditor.class              LayoutManagerEditor.class
RectangleEditor.class            StringArrayEditor.class
StringEditor.class               SymPropertyEditorSupport.class
URLArrayEditor.class             URLEditor.class
UseParentsBackgroundEditor.class UseParentsFontEditor.class
UseParentsForegroundEditor.class
             19 File(s)         18,547 bytes


   Directory of
D:\Inetpub\wwwroot\time\classes\symantec\itools\vcafe\beanhelp\simpleresource

[.]                              [..]
Handler.class                    SimpleResourceConnection.class
             4 File(s)          3,042 bytes


   Directory of D:\Inetpub\wwwroot\time\classes\symantec\itools\vcafe\commtool

[.]                              [..]
Memory.class                     MeterTool$1.class
MeterTool$SymItem.class          MeterTool$SymWindow.class
MeterTool.class
             7 File(s)          7,771 bytes


   Directory of D:\Inetpub\wwwroot\time\classes\symantec\tools

[.]          [..]           [debug]    [java]     [javac]     [ttydebug]
             6 File(s)                  0 bytes


   Directory of D:\Inetpub\wwwroot\time\classes\symantec\tools\debug

[.]
[..]
Agent.class
AgentConstants.class
AgentIn.class
AgentOutputStream.class
BreakpointHandler.class
BreakpointQueue.class
BreakpointSet.class
DebuggerCallback.class
EmptyApp.class
ExecReRouter.class
ExpressionOutOfContextException.class
ExpressionWrongNumberLocalVarsException.class
Field.class
InvalidPCException.class
LineNumber.class
LocalCommInputStream.class
LocalCommOutputStream.class
LocalCommStream.class
```

```
LocalVariable.class
MainThread.class
NoSessionException.class
NoSuchFieldException.class
NoSuchLineNumberException.class
RemoteAgent.class
RemoteArray.class
RemoteBoolean.class
RemoteByte.class
RemoteChar.class
RemoteClass.class
RemoteDebugger.class
RemoteDouble.class
RemoteField.class
RemoteFloat.class
RemoteInt.class
RemoteLong.class
RemoteObject.class
RemoteShort.class
RemoteStackFrame.class
RemoteStackVariable.class
RemoteString.class
RemoteThread.class
RemoteThreadGroup.class
RemoteValue.class
ResponseStream.class
StackFrame.class
SymDbgExec.class
SymDbgExprFindContext.class
SymDbgExprHandler.class
SymDbgExprResult.class
SymDbgLoadExpr.class
SymDbgNotify.class
SymDbgProcess.class
ThreadList.class
              55 File(s)          136,888 bytes


 Directory of D:\Inetpub\wwwroot\time\classes\symantec\tools\java


[.]                          [..]
VEDbgJavaDataRec.class       VEDbgJavaDebug.class
VEDbgJavaIntf.class          VEDbgJavaKernel.class
VEDbgJavaMsgRouter.class     VEDbgJavaRemoteApp.class
VEDbgJavaResource.class      VEDbgJavaResource_ja.class
VEDbgJavaRuntimeCmd.class
              11 File(s)          39,759 bytes


 Directory of D:\Inetpub\wwwroot\time\classes\symantec\tools\javac


[.]                      [..]                    BatchEnvironment.class
BatchParser.class        ErrorConsumer.class     ErrorMessage.class
JavaIDDE.class           Main.class
              8 File(s)          33,103 bytes


 Directory of D:\Inetpub\wwwroot\time\classes\symantec\tools\ttydebug


[.]          [..]          TTY.class
```

309

```
              3 File(s)            28,429 bytes

 Directory of D:\Inetpub\wwwroot\_private

[.]   [..]
              2 File(s)                 0 bytes

 Directory of D:\Inetpub\wwwroot\_vti_bin

fpcount.exe    shtml.dll
              2 File(s)           121,616 bytes

 Directory of D:\Inetpub\wwwroot\_vti_bin\_vti_adm

admin.dll
              1 File(s)            15,120 bytes

 Directory of D:\Inetpub\wwwroot\_vti_bin\_vti_aut

author.dll    dvwssr.dll
              2 File(s)            21,536 bytes

 Directory of D:\Inetpub\wwwroot\_vti_cnf

AdvertisementRotator.asp       Advertisements.txt
default.asp                    index.asp
ManipulatingTextStreams.asp    normal.asp
NumberError.html               ThankYou.html
WebBrowserCapabilities.asp
              9 File(s)             5,048 bytes

 Directory of D:\Inetpub\wwwroot\_vti_pvt

access.cnf    botinfs.cnf    bots.cnf      deptodoc.btr    doctodep.btr
linkinfo.cnf  service.cnf    service.lck   services.cnf    svcacl.cnf
uniqperm.cnf  writeto.cnf
              12 File(s)            24,551 bytes

     Total Files Listed:
          3728 File(s)      15,812,582 bytes
                           925,600,768 bytes free
```

ADMINISTRATIVE MESSAGE
ROUTINE
R 211930Z OCT 98 ZYB MIN PSN 498013S25
FM SECNAV WASHINGTON DC//SN//
TO ALNAV
UNCLAS //N01500//
ALNAV 084/98
MSGID/GENADMIN/SECNAV//
SUBJ/DEPARTMENT OF THE NAVY WORLDWIDE WEB POLICY//
REF/A/DOC/DEPSECDEF/25SEP98/-/NOTAL//
REF/B/DOC/DOD/18JUL97//
REF/C/RMG/CNO/21JUL95//
REF/D/DOC/CNO/18SEP88//
REF/E/DOC/CNO/29AUG95//
REF/F/DOC/USC/1974//
NARR/REF A IS A MEMORANDUM ADDRESSING INFORMATION VULNERABILITY AND
THE WORLDWIDE WEB. REF B IS DOD POLICY FOR ESTABLISHING AND
MAINTAINING A PUBLICLY ACCESSIBLE DOD WEB INFORMATION SERVICE. REF
C, ALCOM 035/95, FORMERLY PROVIDED NAVY GUIDELINES FOR USE OF THE
INTERNET. THIS ALNAV SUPERSEDES REF C. REF D, OPNAVINST 5510.1H,
DEPARTMENT OF THE NAVY INFORMATION SECURITY PROGRAM REGULATION, REF
E, OPNAVINST 3432.1, OPERATIONS SECURITY. REF F, THE PRIVACY ACT OF
1974, 5 U.S.C. SECTION 552A//

RMKS/1. AS STATED IN REF A, SOME INFORMATION ON OUR PUBLICLY
ACCESSIBLE WEB SITES PROVIDES TOO MUCH DETAIL ON NAVAL CAPABILITIES,
INFRASTRUCTURE, PERSONNEL AND OPERATIONAL PROCEDURES. SUCH DETAIL,
WHEN COMBINED WITH INFORMATION FROM OTHER SOURCES, MAY INCREASE THE
VULNERABILITY OF DOD SYSTEMS AND POTENTIALLY BE USED TO THREATEN OR
HARASS NAVAL PERSONNEL AND THEIR FAMILIES.

2. ALL NAVAL COMMANDERS WHO ESTABLISH PUBLICLY ACCESSIBLE WEB SITES
ARE RESPONSIBLE FOR ENSURING THAT THE INFORMATION PUBLISHED ON THEIR
SITES DOES NOT COMPROMISE NATIONAL SECURITY OR PLACE DOD PERSONNEL
AT RISK. THE COMMANDER'S RESPONSIBILITY EXTENDS BEYOND GENERAL
PUBLIC AFFAIRS CONSIDERATIONS REGARDING THE RELEASE OF INFORMATION
INTO THE REALM OF OPERATIONAL SECURITY AND FORCE PROTECTION.
COMMANDERS MUST APPLY COMPREHENSIVE RISK MANAGEMENT PROCEDURES TO
ENSURE THAT THE CONSIDERABLE MISSION BENEFITS GAINED BY USING THE
WEB ARE CAREFULLY BALANCED AGAINST THE POTENTIAL SECURITY AND
PRIVACY RISKS CREATED BY HAVING AGGREGATED INFORMATION MORE READILY
ACCESSIBLE TO A WORLDWIDE AUDIENCE.

3. DEFINITION - PUBLICLY ACCESSIBLE WORLDWIDE WEB PAGE: ANY
WORLDWIDE WEB PAGE, DIRECTLY OR INDIRECTLY, CONNECTED TO THE
INTERNET/NIPRNET, TO WHICH EXTERNAL ACCESS IS NOT CONTROLLED VIA AN
AUTHENTICATION MECHANISM; SUCH AS USER ID AND PASSWORD. SITES WHICH
SOLELY EMPLOY CUSTOMER ADDRESS FILTERING (I.E., RESTRICTING ACCESS
TO .MIL ADDRESSES) ARE NOT CONSIDERED ADEQUATE TO PRECLUDE PUBLIC
ACCESSIBILITY.

4. ACTION. ALL NAVAL COMMANDERS WHO HAVE ESTABLISHED PUBLICLY
ACCESSIBLE WEB SITES SHALL IMMEDIATELY REVIEW THE CONTENT OF THOSE
SITES AND REMOVE THE FOLLOWING INFORMATION:

A. PLANS OR LESSONS LEARNED WHICH WOULD REVEAL SENSITIVE MILITARY
OPERATIONS, EXERCISES, OR VULNERABILITIES.
B. REFERENCE TO ANY INFORMATION THAT WOULD REVEAL SENSITIVE
MOVEMENTS OF MILITARY ASSETS OR THE LOCATION OF UNITS, INSTALLATIONS,
OR PERSONNEL WHERE UNCERTAINTY REGARDING LOCATION IS AN ELEMENT OF
THE SECURITY OF THE MILITARY PLAN OR PROGRAM.
C. ALL PERSONAL INFORMATION IN THE FOLLOWING CATEGORIES ABOUT U.S.
CITIZENS, DOD EMPLOYEES AND MILITARY PERSONNEL: 1)SOCIAL SECURITY
ACCOUNT NUMBERS; 2)DATES OF BIRTH; 3)HOME ADDRESSES AND 4)TELEPHONE
NUMBERS OTHER THAN PHONE NUMBERS OF DUTY OFFICES WHICH ARE
APPROPRIATELY MADE AVAILABLE TO THE GENERAL PUBLIC. IN ADDITION,
REMOVE NAMES, LOCATIONS AND ANY OTHER IDENTIFYING INFORMATION ABOUT
FAMILY MEMBERS OF DOD EMPLOYEES AND MILITARY PERSONNEL.

5. IF DETERMINED THAT THE IMMEDIATE REMOVAL OF INFORMATION WOULD
ADVERSELY IMPACT ESSENTIAL MISSION ACCOMPLISHMENT WAIVERS MUST BE
REQUESTED VIA CHAIN OF COMMAND. ALL NAVAL COMMANDERS WILL REPORT
THROUGH THEIR IMMEDIATE SUPERIOR VIA THE CHAIN OF COMMAND (ISIC) TO
SECOND ECHELON COMMANDERS WHEN THIS ACTION HAS BEEN COMPLETE. ALL
SECOND ECHELON COMMANDERS WILL REPORT TO DON CIO UPON COMPLETION OF
THIS TASKING BY THEIR CLAIMANCY NLT 15NOV98. ALL USMC UNITS WILL
REPORT THROUGH AC/S C4I POC.

6. DURING THIS PROCESS, COMMANDERS/COMMANDING OFFICERS WILL EVALUATE
THE SENSITIVITY OF TECHNOLOGICAL DATA ON THEIR WEB SITES. THESE
ASSESSMENTS WILL ADDRESS THE EXTENT THAT SUCH INFORMATION, WHEN
COMPILED WITH OTHER UNCLASSIFIED INFORMATION, REVEALS AN ADDITIONAL
ASSOCIATION OR RELATIONSHIP THAT MEETS THE STANDARDS FOR
CLASSIFICATION UNDER SECTION 1.8 (E) EXECUTIVE ORDER 12958.
RECOMMENDATIONS ADDRESSING THIS ISSUE WILL BE INCLUDED IN REPORTS TO
RESPECTIVE ISICS AND DON CIO.

7. A DOD TASK FORCE WILL DEVELOP POLICY AND PROCEDURAL GUIDANCE
RELATED THAT ADDRESSES THE OPERATIONAL, PUBLIC AFFAIRS, ACQUISITION,
TECHNOLOGY, PRIVACY, LEGAL AND SECURITY ISSUES RELATED TO THE USE OF
DOD WEB SITES. THIS GUIDANCE WILL BE PROMULGATED IN APPROXIMATELY
60 DAYS. DESIGNATED DON INDIVIDUALS WILL PARTICIPATE IN THIS EFFORT
AS WELL AS DEVELOP TAILORED AMPLIFYING POLICY FOR DON. IN THE
INTERIM, NAVAL COMMANDERS WILL ENSURE THAT THEIR WEB SITES COMPLY
WITH THE GUIDANCE PROVIDED IN THIS MESSAGE AND IN REF B.

8. INTERIM POLICY. NAVAL COMMANDERS SHALL REVIEW THEIR PUBLICLY
ACCESSIBLE WEB SITES TO ENSURE COMPLIANCE WITH THE FOLLOWING:
A. ALL INFORMATION SYSTEMS WITH PUBLICLY ACCESSIBLE SERVERS WILL BE
CERTIFIED, ACCREDITED AND RECEIVE A FORMAL AUTHORIZATION TO OPERATE
BY THE DESIGNATED APPROVING AUTHORITY (DAA). A NETWORK RISK ANALYSIS
MUST BE CONDUCTED AS PART OF THE OVERALL NETWORK SECURITY PLAN TO
DETERMINE THE APPROPRIATE LEVEL OF SECURITY. DON WAN/LAN SYSTEMS
SECURITY ACCREDITATIONS MUST BE UPDATED TO REFLECT THE ADDITION
OF, OR EXISTENCE OF, A WEB SERVER OR OTHER INTERNET INFORMATION

312

SERVER.
B. COMMANDING OFFICERS SHALL APPOINT A PRIMARY AND AN ALTERNATE
WEBMASTER, IN WRITING, AS THE COMMAND'S WEBMASTER. THE PRIMARY
WEBMASTER OVERSEES THE COMMAND'S WEB SITE AND ENSURES COMPLIANCE
WITH CURRENT DIRECTIVES. EACH HOME WEB PAGE WILL HAVE A DESIGNATED
AUTHOR OR MAINTAINER, WHO WILL BE RESPONSIBLE FOR THE CONTENT
AND APPEARANCE OF THAT WEB PAGE. THIS INDIVIDUAL'S ORGANIZATIONAL
CODE, DATE OF LAST REVISION, AND AN "APPROVED BY" STATEMENT (IAW
SUBPARA E BELOW) SHALL BE INCLUDED IN THE SOURCE CODE FOR EACH WEB
HOME PAGE. SINCE THE INTERNET IS OPEN AND LEGALLY ACCESSED BY THE
WORLDWIDE PUBLIC, INFORMATION PRESENTED IN PUBLICLY ACCESSIBLE WEB
SITES REFLECTS ON THE DEPARTMENT OF THE NAVY'S PROFESSIONAL STANDARDS
AND CREDIBILITY. REGARDLESS OF HOW OR BY WHOM THESE PAGES ARE
ACTUALLY DEVELOPED, THE APPEARANCE, AND THE ACCURACY, CURRENCY AND
RELEVANCE OF THIS INFORMATION REFLECTS DIRECTLY, OR INDIRECTLY, ON
THE DEPARTMENT OF THE NAVY'S IMAGE. INFORMATION RESIDING ON A SERVER
WITH A .MIL DOMAIN, MAY BE INTERPRETED BY THE WORLDWIDE PUBLIC,
INCLUDING THE AMERICAN TAXPAYER AND MEDIA, AS REFLECTING OFFICIAL
DON OR DOD POLICIES OR POSITIONS. THERE IS NO SUCH THING AS A
PERSONAL OR UNOFFICIAL WEB PAGE ON A ".MIL" SERVER. THESE SERVERS
AND THE INFORMATION THEY CONTAIN SHALL BE USED ONLY FOR OFFICIAL
BUSINESS AND IN AN OFFICIAL CAPACITY. DETAILED GUIDANCE REGARDING
NON-PUBLIC WEB SITES WILL BE PROMULGATED IN FORTHCOMING SECNAV
POLICY.
C. PUBLICLY AVAILABLE INFORMATION WILL NOT INCLUDE CLASSIFIED
MATERIAL, INFORMATION THAT IS SENSITIVE IN NATURE, OR INFORMATION
THAT COULD ENABLE THE RECIPIENT TO INFER CLASSIFIED INFORMATION.
REQUIREMENTS FOR INFORMATION SECURITY, TO INCLUDE CLASSIFICATION,
MARKING, SAFEGUARDING, TRANSMITTING AND DECLASSIFYING, AS
PROMULGATED IN REF D WILL BE FOLLOWED REGARDING ALL DATA RESIDENT
ON INFORMATION SYSTEMS.
D. PUBLICLY AVAILABLE INFORMATION WILL NOT VIOLATE PERSONAL PRIVACY
OR THE REQUIREMENTS OF THE PRIVACY ACT (REF F). PUBLICLY
AVAILABLE INFORMATION WILL NOT VIOLATE DOD POLICY THAT PROTECTS THE
DISCLOSURE OF NAMES AND DUTY STATION ADDRESSESS OF INDIVIDUALS WHO
ARE STATIONED OVERSEAS, ROUTINELY DEPLOYABLE OR IN SENSITIVE UNITS,
WITH EXCEPTION OF FLAG OFFICERS AND PUBLIC AFFAIRS OFFICIALS. ALL
COMMANDS WILL ENSURE THAT PHOTOGRAPHS, E-MAIL ADDRESSES,
ORGANIZATIONAL CHARTS THAT LIST NAMES, AND OTHER BIOGRAPHICAL DATA OF
INDIVIDUALS ARE NOT PLACED ON PUBLICLY AVAILABLE WEB SITES. NEITHER
PUBLICLY AVAILABLE WEB SITES, NOR NON-PUBLIC DON WEB SITES WILL
CONTAIN INFORMATION WHICH WOULD VIOLATE THE PRIVACY ACT.
E. THE PLACEMENT OF ELECTRONIC READING ROOM MATERIALS ON A COMMAND'S
WEB SITE MUST BE CLEARED FOR THIRD PARTY DISCLOSURE UNDER THE
PROVISIONS OF THE FREEDOM OF INFORMATION ACT (FOIA).
F. PUBLICLY AVAILABLE INFORMATION WILL BE CLEARED THROUGH THE
COMMAND'S OR ISIC'S PUBLIC AFFAIRS OFFICER (PAO), WHO WILL ENSURE
THE INFORMATION FOLLOWS ALL CURRENT PUBLIC INFORMATION RELEASE AND
SECURITY DIRECTIVES AND IS APPROVED FOR RELEASE BY THE COMMAND'S
PAO/FOIA OFFICIAL PRIOR TO BEING PLACED ON THE COMMAND'S WEB
SITE. THE PAO WILL REGISTER THE PUBLICLY ACCESSIBLE WEB SITE WITH
THE GOVERNMENT INFORMATION LOCATOR SERVICE (GILS) IAW PARA 4.3 OF
REF B.
G. THE WEB SITE HOME PAGE WILL DISPLAY A TAILORED PRIVACY AND

313

SECURITY NOTICE AS STIPULATED IN PARA 4.4 OF REF B.
H. ALL EXTERNAL HYPERTEXT LINKS MUST ADHERE TO THE GUIDELINES OF
PARA 4.5 OF REF B.

9. PURSUANT TO REF A, DIRECTOR, NAVAL CRIMINAL INVESTIGATIVE SERVICE
(NCIS) WILL COLLABORATE WITH DIRECTOR OF NAVAL INTELLIGENCE (DNI) TO
ENSURE A COMPREHENSIVE, MULTI-DISCIPLINE SECURITY ASSESSMENT IS
CONDUCTED FOR DON WEB SITES WITHIN 3 MONTHS OF PROMULGATION OF THE
FORTHCOMING DOD POLICY (DISCUSSED IN PARA 5 ABOVE). AN ANNUAL
ASSESSMENT OF THESE SITES WILL BE CONDUCTED THEREAFTER.

10. POINTS OF CONTACT REGARDING THIS POLICY ARE THE FOLLOWING:
A. DON CIO: MR. JOSEPH BROGHAMMER, (703) 602-6901.
B. CNO N643: CWO2 THOMAS DELAINE (703) 601-1278.
C. USMC C4I: MS. GILDA MCKINNON (703) 607-5546.
D. CHINFO: ALAN GOLDSTEIN (703) 695-1887.

11. RELEASED BY THE HONORABLE JOHN H. DALTON, SECRETARY OF THE
NAVY.//

BT
NNNN

INITIAL DISTRIBUTION LIST

1. Defense Technical Information Center..2
   8725 John J. Kingman Rd., STE 0944
   Ft. Belvoir, Virginia 22060-6218

2. Dudley Knox Library..2
   Naval Postgraduate School
   411 Dyer Road
   Monterey, California 93943-5101

3. Chairman, Code CS..1
   Naval Postgraduate School
   411 Dyer Road
   Monterey, California 93943-5101

4. Dr. C. Thomas Wu, Code CS/KA..2
   Naval Postgraduate School
   Monterey, California 93943-5100

5. Dr. A. Ciaviarelli..3
   The School of Aviation Safety
   Naval Postgraduate School (Code 10)
   1588 Cunningham RD, RM 301
   Monterey, California 93943-5202

6. Mr. Richard Healing..1
   Department of the Navy Safety and Survivability
   Bldg. 36 Washington Navy Yard
   901 M Street, S.E.
   Washington, D.C. 20374-5028

7. LT Jonathan Held..2
   4736 Oxford Road
   Macon, Georgia 31210

8. CDR Fred Mingo..2
   304 Metz Road
   Seaside, California 93955